

Vysoká škola báňská - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

**Využití akcelerometru v mobilním telefonu k řízení
vzdáleného zařízení**
Remote Control using Cell Phone Accelerometer

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Zadání diplomové práce

Student:

Bc. Petr Rubeš

Studijní program:

N2647 Informační a komunikační technologie

Studijní obor:

2601T013 Telekomunikační technika

Téma:

Využití akcelerometru v mobilním telefonu k řízení vzdáleného zařízení
Remote Control using Cell Phone Accelerometer

Zásady pro vypracování:

Cílem práce je popsat získávání dat z akcelerometru v mobilním telefonu a jejich využití pro řízení vzdáleného zařízení. Dále bude vytvořena mobilní aplikace pro řízení zařízení a popis komunikace mobilního a vzdáleného zařízení.

1. Popis akcelerometru a získávání dat z akcelerometru v mobilním telefonu.
2. Analýza a zpracování získaných dat.
3. Popis dostupných hardwareových prostředků pro komunikaci s mobilním zařízením.
4. Vytvoření mobilní aplikace pro řízení zvoleného vzdáleného zařízení.
5. Otestování vytvořeného zařízení v reálném prostředí.

Seznam doporučené odborné literatury:


- [1] Massimo Banzi. *Getting Started with Arduino*. O'Reilly, 2009. 128 p. ISBN: 978-0-596-15551-3. ISBN 10: 0-596-15551-4.
- [2] Rick Rogers, John Lombardo, Zigurd Mednieks, G. Blake Meike. *Android Application Development: Programming with the Google SDK*. O'Reilly, 2009. 336 p. ISBN-10: 0596521472. ISBN-13: 978-0596521479.
- [3] Simon Monk. *Making Android Accessories with IOIO*. O'Reilly, 2012. 72 p. ISBN-10: 1449323286. ISBN-13: 978-1449323288.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Jan Skapa, Ph.D.**

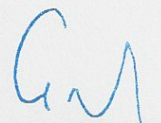
Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2013



prof. RNDr. Vladimír Vašínek, CSc.
vedoucí katedry



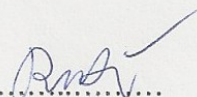


prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne 01.05.2013

..........
podpis studenta

Abstrakt

Cílem práce je nejprve popsat získávání dat z akcelerometru v mobilním zařízení s operačním systémem Android. Data z akcelerometru mají svá specifika a nepřesnosti, proto bude popsáno, jak data upravit a použít k řízení vzdáleného zařízení. Dále bude vytvořen soupis dostupných hardwarových prostředků pro komunikaci s mobilním zařízením, a také popsán způsob komunikace mezi nimi. V neposlední řadě bude práce obsahovat implementaci vlastního programu a elektronického zařízení, na kterém bude demonstrováno řízení pomocí akcelerometru v reálném prostředí.

Klíčová slova

akcelerometr, mobilní telefon, řízení, vzdálené zařízení, Android, Bluetooth, Java, ADK, Arduino, Android IOIO, open source hardware

Abstract

The aim of the thesis is to describe the data acquisition from the accelerometer installed in mobile devices that are running Android operation system. Data from the accelerometer have their own characteristics and inaccuracies, so there will be described how to edit them and use them for controlling remote device. After that, the list of available hardware resources for communication with the mobile device will be created. Way of communication between them will be described too. Finally, the work will include the implementation of the program and the electronic circuit itself, where will be demonstrated how to use accelerometer control in a real environment.

Key words

akcelerometer, mobile phone, controll, remote device, Android, Bluetooth, Java, ADK, Arduino, Android IOIO, open source hardware

Seznam použitých symbolů

Symbol	Jednotky	Význam symbolu
C	F	Kapacita
U	V	Napětí
I	A	Proud
a	m/s ²	Zrychlení
g	m/s ²	Tíhové zrychlení
f	Hz	Frekvence
s	s	Čas
FLOPS	flop/s	Výpočetní výkon

Seznam použitých zkratek

Zkratka	Anglický význam	Český význam
1-Wire	Data and power over a signal line.	Data a napájení po jednom vodiči.
ADK	Accessory Development Kit	Kit umožňující vytvářet příslušenství pro zařízení s OS Adroid
API	Application Programming Interface	Sbírka procedur, funkcí či tříd určité knihovny.
Bluetooth	Standard that allows you to create personal area networks.	Standard umožňující vytvářet osobní síť.
I²C	Inter-Integrated Circuit	Sběrnice propojující integrované obvody.
MEMS	Micro-Electro-Mechanical Systems	Mikro Elektromechanický Systém
OS	Operating system	Operační systém
OSHW	Open Source Hardware	Hardware s otevřenými prostředky
OSI	Open Systems Interconnection	Popisuje komunikaci v počítačových sítích pomocí vrstev.
PWM	Pulse Width Modulation	Pulzně Šířková Modulace
RS232	Serial port	Sériový port
SPI	Serial Peripheral Interface Bus	Sériové periferní rozhraní
USB	Universal Serial Bus	Univerzální sériová sběrnice
WiFi	Wireless Fidelity	Technologie pro připojení zařízení do počítačové sítě.
X10	Home automation protocol. Uses power line wiring for signaling and control.	Protokol pro domácí automatizaci. Používá stávající elektro instalaci.

Seznam použitých termínů

Termín	Význam termínu
Akcelerometr	Čidlo zrychlení, dnes zastoupeno ve většině mobilních zařízení ve všech třech osách pohybu.
Firmware	Malý program, který řídí mikroprocesory. Je nahrán v jejich vnitřní paměti.
Gyroskop	Detekuje rotační pohyb kolem své osy.
Listener	Posluchač dané události. Při změně naslouchaného prostředku vyvolá námi definovanou reakci.
Mobilní zařízení	Přenosné multifunkční elektronické zařízení. V této práci zejména mobilní telefon nebo tablet.
Vzdálené zařízení	Zařízení ve vzdálenosti několika metrů od řídicího bodu. Jeho vzdálenost se může měnit, proto je pro komunikaci většinou použita bezdrátová technologie.

Obsah

1	Úvod	1
2	Akcelerometr.....	2
2.1	Použití akcelerometru v dnešních mobilních telefonech.....	4
2.2	Přístup k datům v OS Android	5
2.3	Metodika testování	6
2.4	Linearita	8
2.5	Hladinový test	10
2.6	Reakce na rychlé zastavení	12
2.7	Rychlost odečítání.....	14
2.8	Úprava reálného signálu.....	16
3	Vhodné hardwarové prostředky	18
3.1	Umístění.....	18
3.1.1	Obvodová část na straně telefonu.....	19
3.1.2	Obvodová část na straně ovládaného zařízení.....	20
3.1.3	Obvodová část uprostřed.....	20
3.1.4	Obvodová část na obou stranách	20
3.2	Komunikační protokoly	21
3.2.1	WiFi.....	21
3.2.2	Bluetooth	22

3.3	Pohony a spínací prvky	22
3.3.1	Motor.....	23
3.3.2	Bezkartáčkový motor	23
3.3.3	Krokový motor	24
3.3.4	Servomotor	25
3.3.5	Relé	25
3.3.6	Unipolární tranzistor	26
3.4	Open Source Hardware	26
3.4.1	Arduino	27
3.4.2	Android IOIO	28
3.4.3	Arduino klon a další	30
4	Vlastní řešení.....	32
4.1	Obvodová řešení.....	32
4.1.1	Použití stávajícího ovladače	32
4.1.2	Řídící obvod na straně vzdáleného zařízení	34
4.1.3	Univerzální modelářské řešení	36
4.2	Řídící aplikace.....	39
4.3	Testování	42
5	Shrnutí a závěr	43
	Citovaná literatura.....	44
	Seznam obrázků	45
	Seznam tabulek	47
	Seznam příloh	48
	Přílohy.....	49

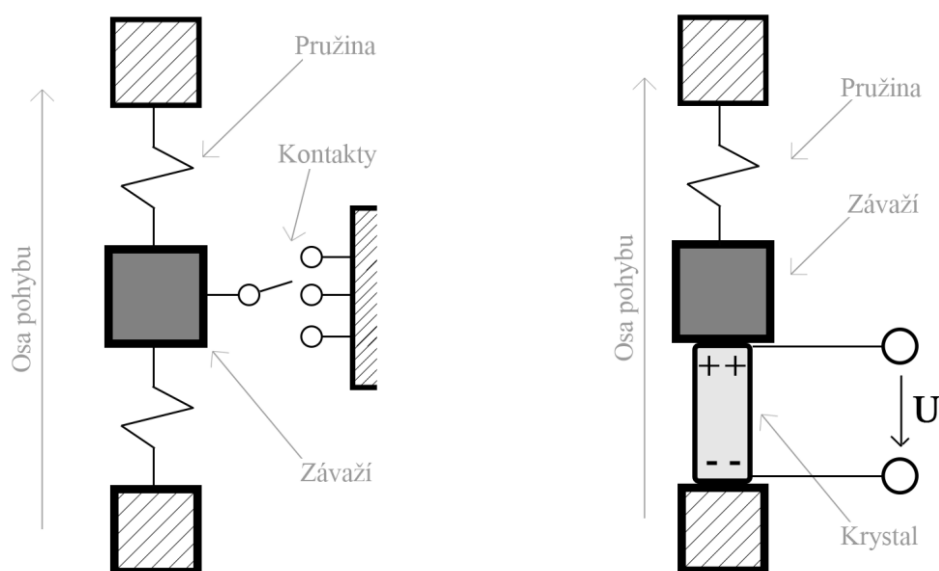
1 Úvod

Čím dál více lidí vlastní „chytrý mobilní telefon“ a další mobilní zařízení jako jsou například tablety nebo notebooky. Díky konkurenčnímu boji výrobců se do těchto zařízení dostávají výkonnější komponenty a senzory. Lze je tedy dobře použít pro různé aplikace, ke kterým nejsou tato zařízení primárně předurčena (mobilní telefon/telefonování). Výrobci ovšem tlačí na cenu, a proto se do zařízení nedostanou vždy ty nejvyšší komponenty. Tato práce je výhradně zaměřena na akcelerometr, který je zastoupen v drtivé většině dnešních mobilních telefonů a tabletů. Ne v každém zařízení je však akcelerometr identický. Různé typy většinou vykazují rozdílné chování. Díky tomuto senzoru může uživatel změnou polohy zařízení interagovat s programy telefonu. Tento druh ovládání však můžeme přenést do reálného prostředí, tedy k řízení ostatních zařízení. Čistá data z akcelerometru nejsou vždy vhodná k ovládání reálných zařízení. Mají svá specifika, která budou popsána a vhodně znázorněna níže. Pokud chceme tato data použít pro řízení, musíme je vhodně upravit. Dalším krokem bude vytvořit a popsat dostupné hardwarové prostředky pro bezdrátovou výměnu řídicích informací mezi mobilním a vzdáleným zařízením. V práci bude na vlastním modelovém příkladu demonstrováno, jak ovládat různé druhy motorů, servomechanizmů a spínacích obvodů.

2 Akcelerometr

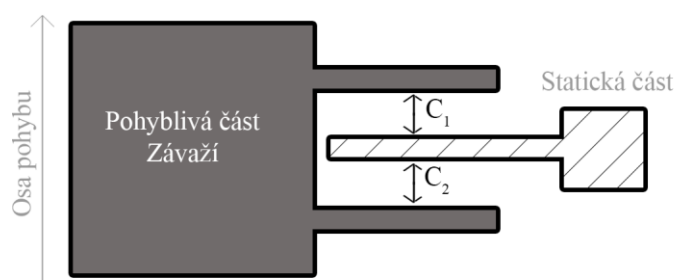
Původní konstrukce akcelerometrů vycházela z klasických mechanických prvků jako jsou například pružiny. Jednoduchý akcelerometr je znázorněn na obrázku 2.1. Tento akcelerometr měří zrychlení v jedné ose pomocí pružiny, závaží a stupnice. Pokud bereme stupnici jako mechanické kontakty, můžeme jednoduše sledovat polohu závaží. Rozlišení akcelerometru je tedy úměrně závislé na počtu kontaktů, tedy i velikosti celé součástky. Pro sledování všech tří os pohybu, je nutné použít tento akcelerometr třikrát. Díky jeho velikosti není vhodná integrace do mobilního zařízení, které se snaží být co nejmenší.

Dalším ve vývojové řadě byl piezoelektrický akcelerometr (obr. 2.1). Tato součástka je založena na piezoelektrickém krystalu, kde vyvíjením tlaku na tento krystal vzniká napětí. Odečítáním tohoto napětí získáme aktuální hodnotu zrychlení. Výstupní rozsah již není závislý na počtu kontaktů na stupnici, ale je popsán rozsahem, do kterého výstupní hodnota spadá. Jelikož je výstupní napětí velice malé, musí být zesíleno na měřitelnou úroveň. Přes jeho výhody, jako je malá velikost, velká frekvenční odezva a generování vlastního napětí, je tento typ akcelerometru však velice náchylný na vlhkost a zejména na teplotu. Dalším parametrem, který není vhodný pro sledování náklonu je nízký výstupní rozsah. Proto se v dnešní době tyto piezoelektrické akcelerometry využívají například k detekci vibrací [1].



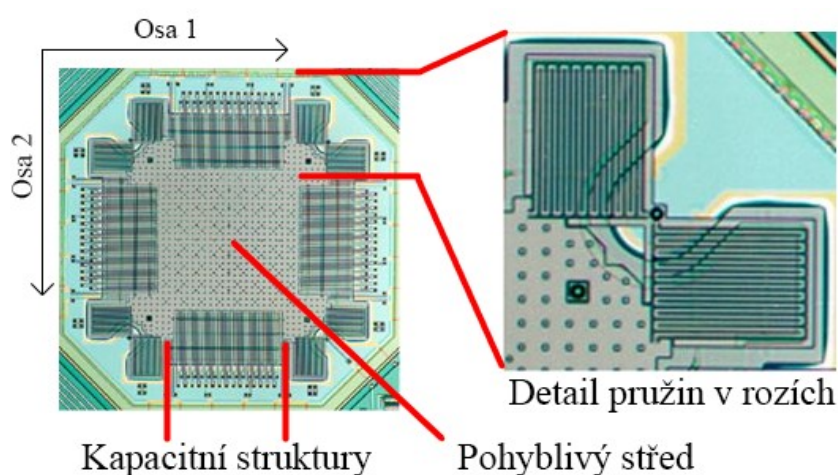
Obrázek 2.1: Akcelerometr z mechanických částí vlevo a akcelerometr s krystalem vpravo

Výrobní technologie MEMS (Micro-Electro-Mechanical Systems) umožňuje vytvářet mechanické křemíkové struktury o velikosti řádů mikrometrů. Díky těmto strukturám mohou obvody dnešních akcelerometrů nabývat velikosti přibližně 3x3x2mm. Nejčastěji je použita kapacitní MEMS struktura, která je popsána na obrázku 2.2, kde dochází ke změně vzdálenosti, tedy i kapacity mezi dvěma statickými a jednou pohyblivou částí [2]. Dvě osy jsou většinou řešeny jednou čtvercovou plochou s mikro pružinami v rozích. Třetí osa je řešena samostatnou strukturou, kde pro zvýšení senzitivity je použito více kapacitních částí (obr. 2.3).



Obrázek 2.2: Kapacitní část MEMS akcelerometru

Dobrým příkladem je reálná fotografie dvouosého akcelerometru ADXL202 pod mikroskopem (obr. 2.3). Na tomto obrázku je dobře viditelná čtvercová struktura s pružinami v rozích. Na stranách čtverce jsou kapacitní části (obr. 2.2), samotná čtvercová struktura slouží jako závaží. Díky vysoké integraci a nízké výrobní ceně jsou akcelerometry vyrobené technologií MEMS osazovány do všech mobilních zařízení dnešní doby.



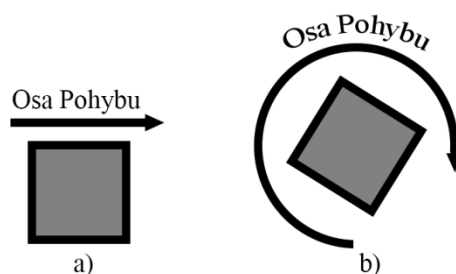
Obrázek 2.3: Akcelerometr ADXL202 pod mikroskopem [3]

2.1 Použití akcelerometru v dnešních mobilních telefonech

Poprvé se dostal akcelerometr do prvních „chytrých“ mobilních telefonů s obdélníkovým displayem, zde sloužil pro plynulé překlápění obrazu mezi režimy zobrazení na výšku a na šířku. Uživatel používá vertikální zobrazení například pro prohlížení kontaktů, vytáčení čísel nebo procházení menu. Pro sledování videa nebo procházení obrázků je vhodnější horizontální zobrazení. Pokud chce uživatel přecházet mezi režimy, jednoduše překllopí telefon a obraz se překllopí také. Tuto funkci zajistí jednoduchý jednoosý akcelerometr.

Postupem času našel akcelerometr uplatnění na poli zábavy. Tříosý akcelerometr nabídl možnost ovládání her v podobě různých labyrintů, emulace volantu v závodních hrách a interakce grafického rozhraní na pohyb uživatele. Dnes akcelerometr plní mnoho podpůrných funkcí o kterých uživatel ani nemusí mít ponětí. Mezi tyto funkce patří například ztlumení vyzvánění zatřepáním telefonu, při zvednutém hovoru lze překlopením zapnout hlasitý odposlech, přiložením k uchu se zamkne display a vrácením do původní polohy se znovu odemkne, pokud je mobil v kapse je zesíleno vyzvánění. Stále více přibývají aplikace, které vyhodnocují jednotlivé vzorky akcelerometru jako je například krokoměr.

Akcelerometr je v zařízení osazen samostatně jako integrovaný obvod na desce plošných spojů mobilního zařízení nebo v kombinaci s gyroskopem v jednom pouzdře. Dnes mnoho lidí chybně zaměňuje název akcelerometr gyroskopem, někdy jsou akcelerometr a gyroskop chybně považovány za stejnou součástku. Akcelerometr však snímá pohyb ve směru své osy a gyroskop snímá rotační pohyb kolem své osy (obr. 2.4). Původní gyroskopy využívaly rotačního závaží jako je setrvačnick, který udržoval rovinu vůči okolnímu prostředí. Proto se využívá v ponorkách a letadlech jako umělý horizont. Gyroskopy v mobilních zařízeních používají MEMS struktury podobně jako akcelerometry.



Obrázek 2.4: a) Pohyb měřený akcelerometrem b) Gyroskopický pohyb

2.2 Přístup k datům v OS Android

Pokud chceme používat senzory v systému Android použijeme API `android.hardware`, které nám umožní přístup k hardwarovým prostředkům, jako jsou senzory a kamera. Nás budou zajímat třídy `Sensor`, `SensorEvent`, `SensorEventListener` a `SensorManager`, které se zde nacházejí. Třída `SensorManager` umožňuje přístup k sensorům. Třída `Sensor` reprezentuje určitý senzor, my použijeme `Sensor.TYPE_ACCELEROMETER`. Výpis všech sensorů se kterými může tato třída pracovat nalezneme v příloze A. Pro výpis všech sensorů obsažených v našem zařízení můžeme použít metodu `getSensorList()` třídy `SensorManager`. `SensorEvent` reprezentuje událost třídy `Sensor` a uchovává načtenou hodnotu. Poslední použitou třídou je `SensorEventListener`. Ta nám umožní vytvořit listener, který bude reagovat na změnu hodnoty akcelerometru a volat metodu `onSensorChanged()`. Nejprve tedy importujeme potřebné třídy.

```
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
```

Při inicializaci programu vytvoříme `SensorManager` deklarovaný globálně privátní proměnou `sensorManager`. Dále vytvoříme listener pro senzor v našem případě `Sensor.TYPE_ACCELEROMETER`. Třída `SensorManager` umožňuje nastavit zpoždění, tedy rychlost načítání hodnot. Jejich výčet je v příloze B.

```
sensorManager=(SensorManager) getSystemService(SENSOR_SERVICE);
sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER),
    SensorManager.SENSOR_DELAY_NORMAL);
```

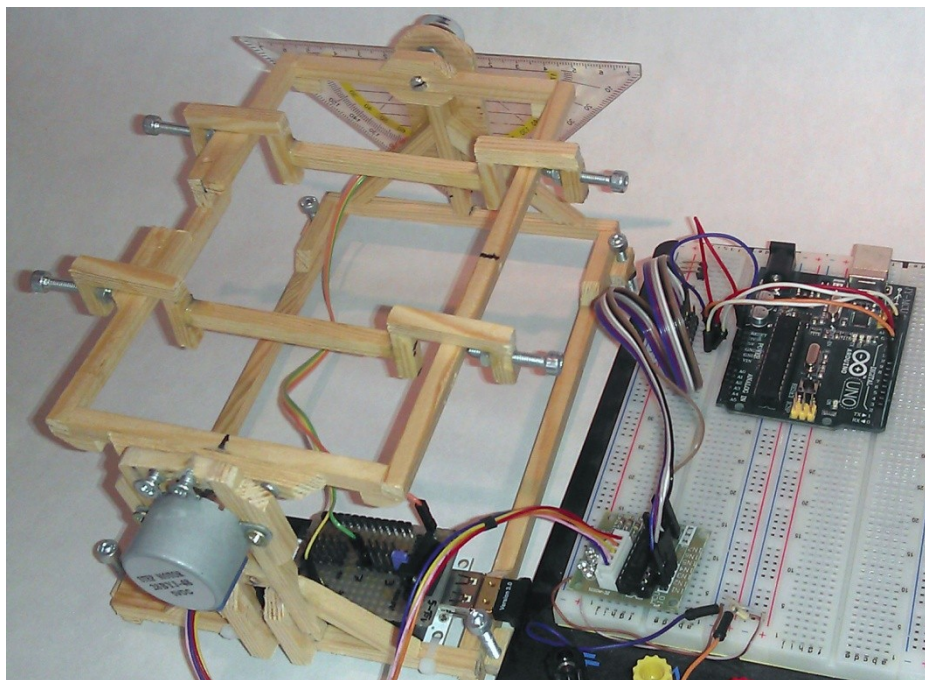
Poslední částí je metoda volaná listenerem při změně hodnoty akcelerometru. V této metodě jsou načtené hodnoty akcelerometru pro všechny osy načteny do proměnných typu `float`.

```
public void onSensorChanged(SensorEvent event){
    if(event.sensor.getType()==Sensor.TYPE_ACCELEROMETER){
        float x=event.values[0];
        float y=event.values[1];
        float z=event.values[2];
    }
}
```


2.3 Metodika testování

Pro testování akcelerometru v telefonu byl pro tuto práci vytvořen speciální přípravek, který pohybuje telefonem v osách (obr. 2.5). Původní myšlenka byla použít samotný telefon pro řízení přípravku, aby se mohl telefon sám otestovat. Z důvodu vyčerpání výpočetní jednotky pomalejších telefonů je pro řízení přípravku použit externí mikroprocesor. Proto není procesor telefonu vyčerpán ovládáním přípravku a může se tak věnovat ukládání hodnot načtených z akcelerometru. První verze používala dva servo pohony pro pohyb v osách, u dalších verzí byl jeden odstraněn. Jednotlivé osy se tedy testují změnou polohy mobilního telefonu v přípravku. Pro korekci základní rovné pozice můžeme použít výškově nastavitelné rohové podpěry. Před každým testem je vždy znovu nastavena optimální základní pozice pomocí měřidel a vodováhy.

První problém, se kterým se setkáme při vlastním návrhu takového přípravku je volba pohonu. Ten musí být rychlý a zároveň precizní. Proto je přípravek navrhnut tak, aby bylo možné vyměnit rychlý servo pohon za precizní krokový motor v závislosti na prováděném testu. Dalším problémem je najít způsob, jak odečítat aktuální polohu přípravku, která bude porovnávána s polohou načtenou z testovaného akcelerometru. Pro zjištění polohy je použit lineární potenciometr v ose přípravku, který je zapojen jako dělič napětí. Odečítáním tohoto napětí zjistíme aktuální polohu zařízení. Další možností je použít upravený servo pohon, který obsahuje svůj vnitřní potenciometr, ze kterého je pro tyto účely vyveden prostřední pin podobně jako v předchozím případě.



Obrázek 2.5: Přípravek pro testování akcelerometru

Testování probíhalo na dvou telefonech značky HTC detailně popsaných v tabulce 2.1. V HTC Desire (dále jen Desire) nalezneme akcelerometr BMA150 firmy Bosch. Jaký typ je v novějším HTC EVO 3D (dále jen Evo3D) však zjistit nelze. Přesný typ akcelerometru v telefonu Evo3D není známý, ani oficiální asijská hardwarová podpora společnosti HTC prý nemá k této informaci přístup. Známý je pouze výrobce InvenSense. S největší pravděpodobností se jedná o jejich řešení kombinující akcelerometr, gyroskop a kompas v jednom pouzdře. Z uživatelského hlediska však není nutné tento údaj znát. Bereme telefon jako celek, proto budou telefony v práci uváděny pod jejich názvem, tedy Evo3D a Desire.

Tabulka 2.1: Detail testovaných zařízení

Název	Rok výroby	Akcelerometr	Verze Android	Procesor
HTC EVO 3D x515m	2011	neznámý	4.0.3	1,2 GHz dual core
HTC Desire A8181	2010	BMA150	2.2	1 GHz

Prováděné testy jsou voleny v závislosti na použití akcelerometru v mobilním telefonu k řízení vzdáleného mechanického zařízení. To znamená, že můžeme použít metody pro vyhlazení signálu, které by v jiných aplikacích (například měření tepu uživatele) odstranili důležitou část signálu. Mobilním telefonem je v testech pohybováno maximálně pod úhlem 45° na všechny strany. Člověk nemůže zařízením pohodlně otočit pod větším úhlem jednou rukou, jak ve vodorovné, tak ve svislé poloze (pravák doprava, levák doleva). Cílem těchto měření bude najít a opravit všechny nepřesnosti v pohybu po jednotlivých osách. Jedná se zejména o šum v polohách, kdy je zařízení v klidu a vznik špiček při rychlém zastavení. To, jak tyto jevy negativně ovlivňují pohyb mechanických částí bude popsáno u příslušných testů. Dále nás bude zajímat nastavená prodleva (Příloha B) senzoru v závislosti na rychlosti výpočetní jednotky mobilního telefonu.

Katalog akcelerometru uvádí několik dalších parametrů, které ovlivňují přesnost jeho práce. Některé parametry jsou pro nás zajímavé, jako je například linearita výstupu. Jsou zde uvedeny další parametry jako je teplotní závislost, senzitivita nebo nepřesnost uložení MEMS struktury v pouzdře a samotné osazení akcelerometru na desce plošných spojů, které nemusí být v ose telefonu a pod stejným úhlem. Tyto další parametry uživatel a návrhář mobilních aplikací neovlivní. Proto bereme telefon jako celek, ve kterém si nemůžeme vybrat typ akcelerometru. Výsledkem by měla být plynulá reakce ovládaného zařízení na pohyb uživatele, kterým si musí být pohybem jistý.

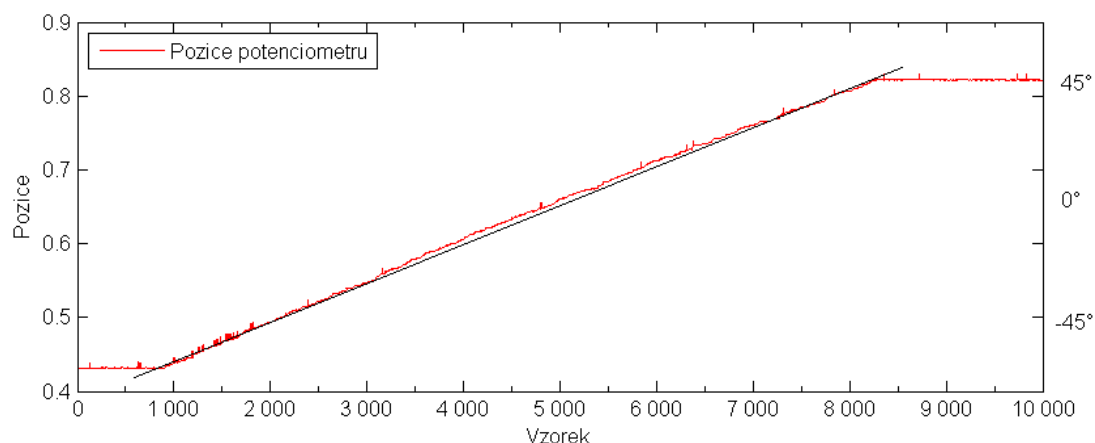
Hodnoty akcelerometru v následujících testech jsou v metrech za sekundu na druhou (m/s^2), tedy přímo v jednotkách zrychlení. Při naklonění telefonu o 90° nabývá hodnota zrychlení přibližně $a = 10 \text{ m/s}^2$, při naklonění o -90° nabývá přibližně $a = -10 \text{ m/s}^2$. Často se setkáme s oblíbenou anglickou jednotkou g-force, tedy česky tíhové zrychlení $g = 9,80665 \text{ m/s}^2$. Vyšších hodnot než $1g$, tedy přibližně 10 m/s^2 , nedosáhneme pouze nakloněním telefonu, ale prudkým škubnutím nebo nárazem. Pokud chceme ovládat zařízení náklonem telefonu, můžeme hodnoty větší než $a = 10 \text{ m/s}^2$ a menší jak $a = -10 \text{ m/s}^2$ zanedbat.

2.4 Linearita

Ačkoli jsou k dostání průmyslové lineární akcelerometry, do telefonů se dostávají levnější nelineární modely. V praxi bude tato odchylka významná při řízení pohonů. Náběh motoru bude při náklonu na jednu stranu jiný, než při náklonu na stranu druhou. Pokud budeme náklonem spínat nějaký spínací prvek, například relé, nebude nelinearita hrát velkou roli.

V zařízeních se systémem android je možná pouze kalibrace základní polohy podle rovného povrchu, kterou může provést sám uživatel. Tato uživatelská kalibrace však není přesná, vybraný povrch nemusí být přesný, navíc většina telefonů nemá rovný zadní kryt. Průmyslové akcelerometry většinou podléhají odborné kalibraci jak od výroby, tak po čas jejich používání. Kalibrace probíhá pomocí úhlově nastavitelných přípravků podobných tomu, který je vytvořen pro tuto práci.

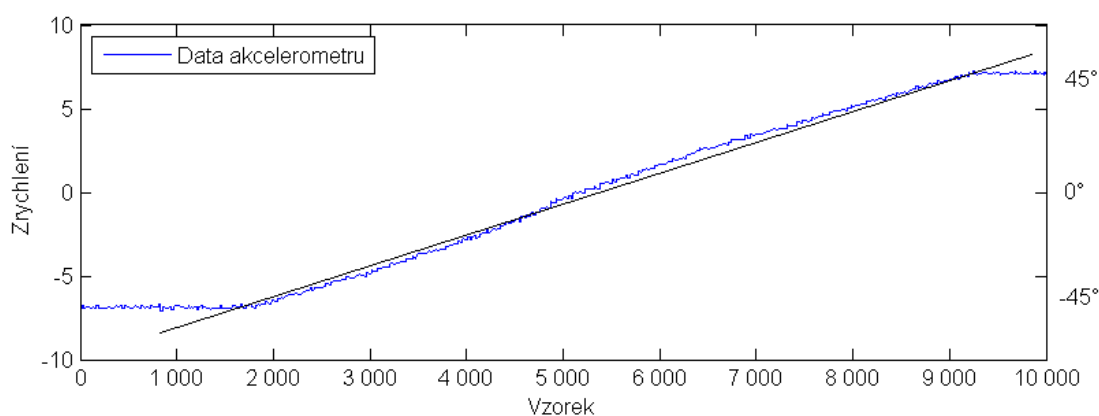
Prvním krokem je ověřit, zda je pohyb přípravku opravdu lineární. Pro tento test byl přípravek osazen krokovým motorem a pohyb byl odečítán lineárním potenciometrem v ose pohybu přípravku. Na výsledném grafu obrázku 2.6 vidíme postupný pohyb přípravku z pozice mínus 45° na opačnou stranu, tedy 45° . Grafem je dále volně vedena přímka, která slouží pro přibližné porovnání strmosti pohybu.



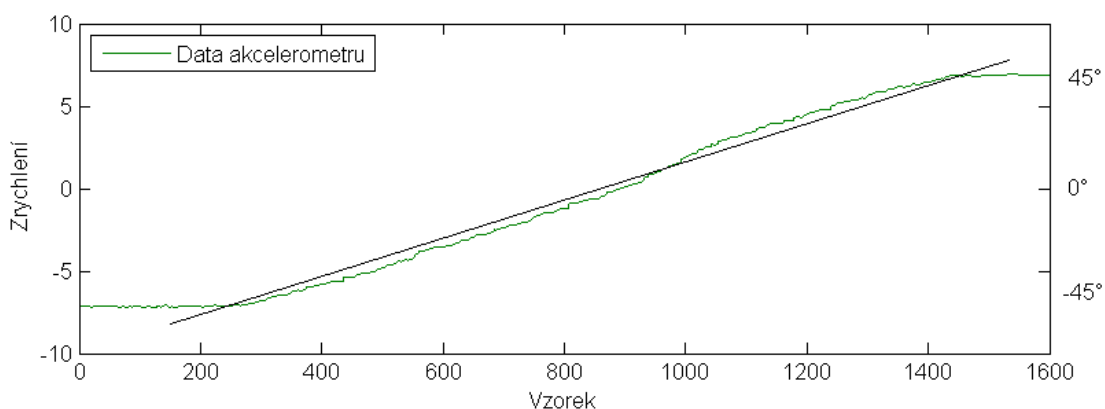
Obrázek 2.6: Linearita přípravku

Z pohledu linearity byla otestována obě zařízení ve všech osách x, y a z. Lehká nelinearita je znát v osách x obou telefonů. Jak Evo3D (obr. 2.7), tak Desire (obr. 2.8) vykazují podobné zkreslení. Obdobné chování vykazují obě zařízení i v osách ostatních, jejich grafické znázornění naleznete v příloze A.

Jelikož samotný Android neumožňuje odbornou kalibraci akcelerometru, nemůžeme nelinearity zcela odstranit. Možné řešení je v závislosti na odměřeném chování kompenzovat výstup akcelerometru ve vlastním programu, který bude data dále zpracovávat. Takové řešení však vyžaduje kalibrační přípravek, který není běžně dostupný. Z pohledu vývojáře musíme s nelinearitami počítat.



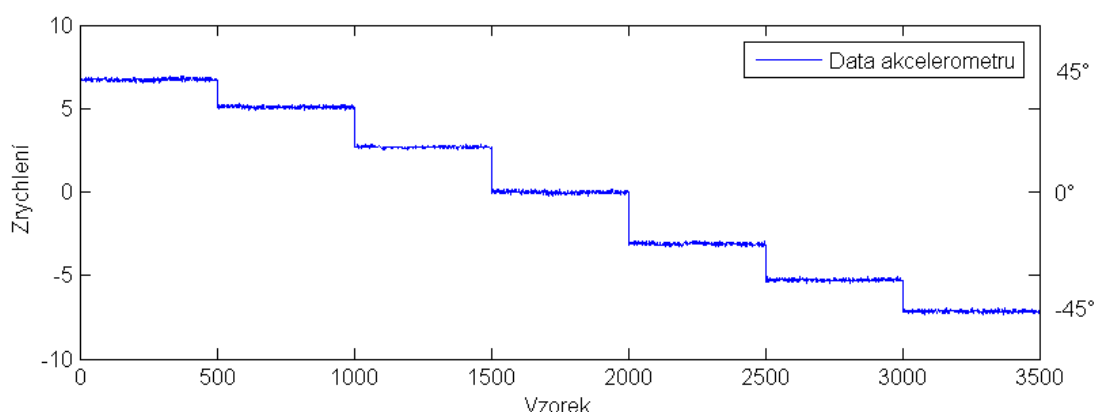
Obrázek 2.7: Linearita Evo3D osa X



Obrázek 2.8: Linearita Desire osa X

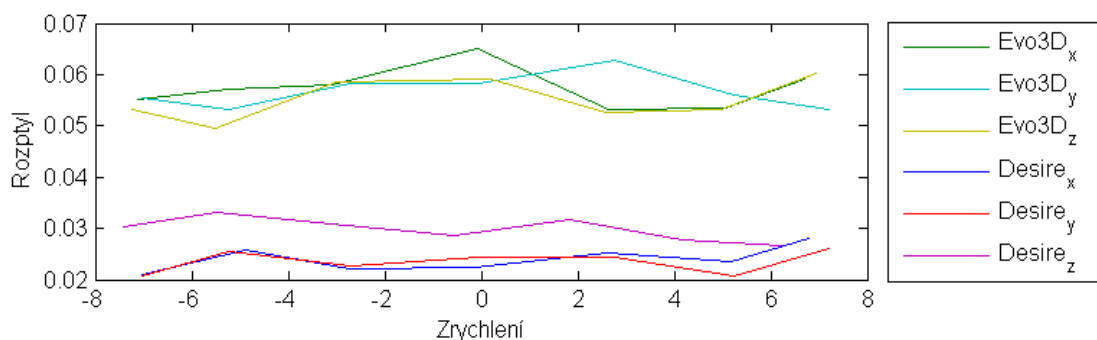
2.5 Hladinový test

Ačkoli akcelerometr obsahuje svůj vlastní vnitřní digitální filtr, který vyhlazuje výstupní hodnoty, jeho výstup stále není vhodný pro řízení určitého typu pohonů. Tento test má prověřit výstupní signál v několika pevných polohách v rozmezí přibližně 45° až minus 45° . Stejně jako v předchozím měření je použit přesnější krokový motor. Celkem v pěti měření je uloženo postupně pět set hodnot pro každou hladinu a osu, výsledná data osy x zařízení Evo3D jsou pro představu znázorněna na obrázku 2.9. Data jsou dále zpracována programem Statgraphics Centurion (30-ti denní zkušební verze 15.1.02).



Obrázek 2.9: Hladinový test osy x zařízení Evo3D

Ze statistik nás zajímá zejména rozptyl hodnot v jednotlivých hladinách, minimální a maximální hodnota. Data jsou uvedena v tabulkách 2.2 pro Evo3D a 2.3 pro Desire. Z naměřených hodnot vidíme přibližně dvakrát větší rozptyl zařízení Evo3D oproti Desire, a to ve všech osách i polohách. Dále chceme zjistit, zda má úhel natočení vliv na rozptyl hodnot. Pomocí naměřených dat můžeme sestavit jednoduchý graf (obr. 2.10), kde vidíme závislost zrychlení a rozptylu hodnot. Evo3D je přesnější okolo natočení 30° , zato Desire vyniká v rovné poloze tedy 0° . Rozdíl mezi hladinami je však zanedbatelný.



Obrázek 2.10: Rozptyly ve všech hladinách

Osa:	x				y				z				
	Střední hodnota	Rozptyl	Min	Max	Střední hodnota	Rozptyl	Min	Max	Střední hodnota	Rozptyl	Min	Max	
Poloha [°]	45	6,6869	0,0590	6,4875	6,8324	7,3359	0,0532	7,1695	7,5144	6,9547	0,0602	6,7270	7,3018
	30	5,0526	0,0534	4,9164	5,2229	5,3814	0,0560	5,2535	5,5600	4,9709	0,0531	4,7727	5,1176
	15	2,6409	0,0531	2,4639	2,8088	2,7707	0,0628	2,6094	2,9159	2,5929	0,0526	2,4351	2,7417
	0	-0,0746	0,0651	-0,2952	0,1263	0,0139	0,0582	-0,1114	0,1952	0,1808	0,0592	-0,1323	0,3658
	-15	-3,1301	0,0581	-3,2842	-2,9393	-2,8091	0,0582	-2,9854	-2,6022	-3,0149	0,0585	-3,1980	-2,8531
	-30	-5,2462	0,0572	-5,3918	-5,0852	-5,2982	0,0531	-5,4379	-5,0930	-5,5074	0,0494	-5,6505	-5,3439
	-45	-7,1221	0,0552	-7,2695	-6,9629	-7,2406	0,0555	-7,4306	-7,0474	-7,2505	0,0533	-7,4132	-7,1066

Tabulka 2.2: Hladinový test Evo3D

Osa:	x				y				z				
	Střední hodnota	Rozptyl	Min	Max	Střední hodnota	Rozptyl	Min	Max	Střední hodnota	Rozptyl	Min	Max	
Poloha [°]	45	6,7812	0,0280	6,7012	6,8919	7,1955	0,0261	7,1234	7,2460	6,2313	0,0265	6,1700	6,3198
	30	5,1681	0,0234	5,1349	5,2166	5,2245	0,0207	5,1757	5,2847	4,1729	0,0279	4,0997	4,2496
	15	2,6070	0,0251	2,5606	2,6832	2,7505	0,0245	2,6832	2,8739	1,8382	0,0316	1,6889	1,9205
	0	-0,1206	0,0225	-0,1907	-0,0817	-0,0591	0,0244	-0,1090	0,0000	-0,5445	0,0286	-0,6946	-0,4631
	-15	-2,7459	0,0220	-2,8330	-2,6832	-2,7161	0,0227	-2,7513	-2,6424	-3,1287	0,0308	-3,4051	-3,0646
	-30	-4,8868	0,0258	-4,9851	-4,7944	-5,2433	0,0256	-5,2847	-5,1757	-5,4839	0,0330	-5,5571	-5,4073
	-45	-7,0223	0,0210	-7,0826	-6,9736	-7,0275	0,0208	-7,0826	-6,9736	-7,4285	0,0303	-7,5048	-7,3550

Tabulka 2.3: Hladinový test Desire

V praxi je velký rozptyl hodnot znatelný při ovládání pohonů, zejména servomotorů. V nejhorším námi naměřeném případě (Evo3D pozice 0° osa x) při překmitu mezi maximální a minimální hodnotou je rozdíl 0,42. Jako vzorový příklad mějme servomotor, který se pohybuje od 0° do 180° , řídicí program budeme přepočítávat zrychlení čtené z akcelerometru $a = 7 \text{ m/s}^2$ až $a = -7 \text{ m/s}^2$ (přibližně náklon 45° až -45°) pro řízení tohoto pohonu. Při změně hodnoty o 0,42 vzniklé rozptylem otočíme servomotorem o $5,4^\circ$. Servomotory jsou velice náchylné na rychlé změny polohy podobné kmitání, kdy zejména trpí převody mezi motorem a hřídelí. Všechny zmíněné hodnoty (tabulka 2.2 a 2.3) jsou měřeny na pevných pozicích řízených přípravkem. Při držení telefonu rukou nedosáhneme tak přesného výstupu akcelerometru. Pro porovnání byla změřena osa x zařízení Evo3D držného rukou, rozptyl činil 0,193 a rozdíl mezi minimem a maximem 2,18. Pokud budeme tuto hodnotu aplikovat na předchozí případ se Servomotorem, dostaneme pohyb větší než 20° . Nutností je tedy provádět další filtraci hodnot z akcelerometru. Jak vhodně upravit získaný signál bude popsáno v dalších kapitolách.

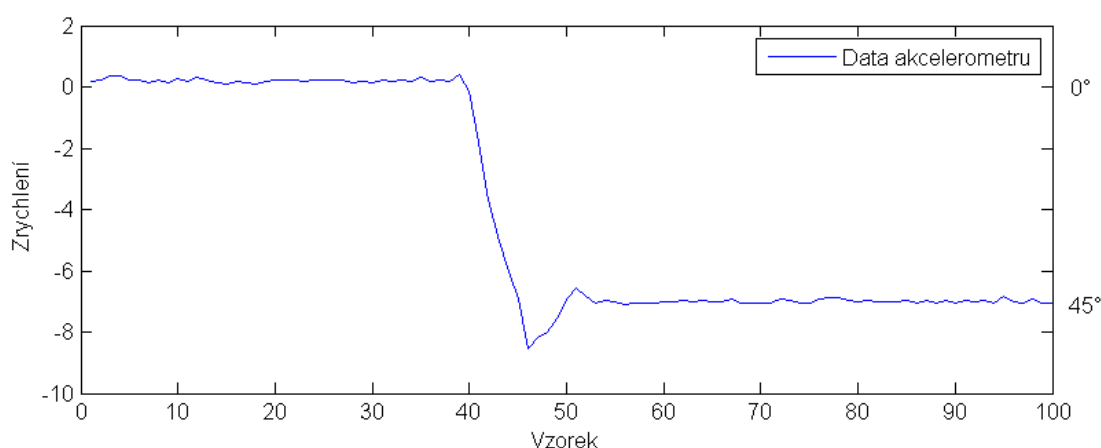
2.6 Reakce na rychlé zastavení

Pokud si má být uživatel pohybem jistý, musí vzdálené zařízení pohotově reagovat na zastavení pohybu mobilního telefonu. V případě rychlého zastavení, do té doby plynulého pohybu, dojde z důvodu zvýšeného přetížení k nárůstu hodnot akcelerometru. Po ustálení pohybu se hodnoty opět vrátí na danou úroveň. Akcelerometr tedy zareaguje správně a vytvořené zrychlení zachytí. Pokud ho však používáme pro řízení vzdáleného zařízení, dojde ke chvilkové odchylce jeho polohy vůči náklonu telefonu. Díky této odchylce uživatel velice znejistí, protože pokud bude chtít rychle zastavit, zařízení chvilkové zrychlí. Dokud nejsou tyto reakce správně ošetřeny programem, může docházet k opotřebení pohonů v důsledku vyvolaného špičkového napětí nebo překračování jejich krajních poloh. Další problém nastane při řízení elektro motorů. Při rychlé změně směru otáčení motor vygeneruje špičkové napětí v opačném směru, podobně jako alternátor. Toto napětí negativně ovlivní ostatní elektroniku napájenou stejným zdrojem a dojde k poklesu napěťových úrovní. Tento pokles může mít za následek restart mikroprocesoru, přerušení komunikace nebo i zničení některých elektronických částí. Tento jev je vhodné řešit i v samotném obvodovém řešení řízeného zařízení.

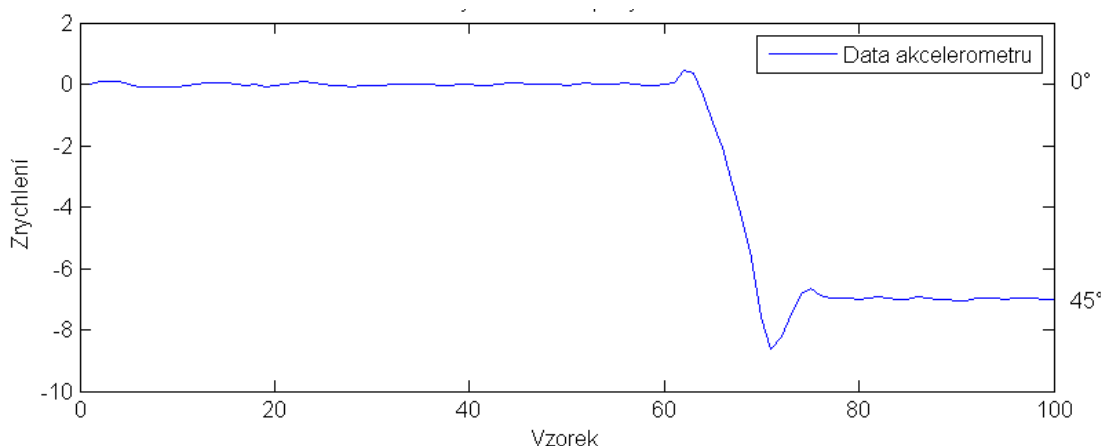
Pro tento test byl použit Servomotor. Jeho pohyb není tak stabilní a přesný jako u motoru krokového, ale je mnohem rychlejší. Oběma testovanými telefony je pohybováno ve všech třech osách. V každé z nich je telefonem ze základní rovné polohy rychle otočeno o 45° . Průběžné hodnoty, které jsou načítány maximální možnou rychlostí, jsou uloženy a dále

zpracovány do grafů. Na obrázku 2.11 je zobrazen průběh testu osy x zařízení Evo3D, který je velice podobný průběhu stejné osy zařízení Desire (obr. 2.12). Na obou je znatelný překmit při prudkém zastavení pohybu. Jeho velikost je přibližně $a = 1 \text{ m/s}^2$. Pokud použijeme velikost tohoto překmitu v našem příkladě se servomotorem z kapitoly 2.5, výsledný nechtěný pohyb bude představovat přibližně 10° .

Výsledné grafy testů ostatních os naleznete v příloze B, průběh os y obou testovaných zařízení je velice podobný osám x. Oproti tomu osy z prokázaly výrazně zhoršené chování, Evo3D vykazuje překmity až $a = 12 \text{ m/s}^2$ a Desire vykazuje špatnou reakci na samotný pohyb. Pro ověření těchto chyb os z bylo provedeno několik dalších měření, které je potvrdily. Toto rozdílné chování osy z oproti x a y je pravděpodobně dáno použitou MEMS strukturou, kdy je pro osy x a y použita jedna čtvercová MEMS struktura a pro z je použita struktura samostatná.



Obrázek 2.11: Reakce na rychlé zastavení osy X zařízení Evo3D



Obrázek 2.12: Reakce na rychlé zastavení osy X zařízení Desire

2.7 Rychlost odečítání

Rychlost získávání dat je důležitá zejména z hlediska plynulosti pohybu. Díky více vzorkům můžeme s daty pracovat více do hloubky. Výsledný čas pro získání jednoho vzorku je dán zejména výpočetním výkonem telefonu (rychlost vykonávání programových instrukcí), vloženým zpožděním systému Android a šířkou pásma akcelerometru. V následujících testech budou popsány závislosti mezi těmito parametry.

Výpočetní výkon byl orientačně naměřen pomocí programu Linpack For Android (ver. 1.2.8, firmy Greene Computing). Tento test vychází ze známé knihovny Linpack, která pomocí výpočtů lineární algebry zjistí hodnotu FLOPS (floating point operations per seconds), označení FLOPS se používá pro určení výpočetního výkonu počítačů. Výsledek testu Evo3D je 56,752 Mflop/s, výsledek Desire je přibližně poloviční, jmenovitě 24,395 Mflop/s. Druhým parametrem je vložené zpoždění systémem android (příloha B). Dle zdrojových kódů je v módu normal zpoždění $s = 200$ ms, ui $s = 66,667$ ms, game $s = 20$ ms a fastest $s = 0$ ms. Maximální rychlost bude v jednotlivých modech: normal $f = 5$ Hz, ui $f = 14,99$ Hz, game $f = 50$ Hz, rychlost módu fastest bude záviset čistě na výpočetním výkonu telefonu a šířce pásma akcelerometru. Posledním parametrem je šířka pásma samotného akcelerometru. Dle katalogu integrovaného obvodu BMA150, který je součástí zařízení Desire, šířka pásma závisí zejména na nastavení digitálního filtru. Tato hodnota je nastavena od výroby telefonu a nelze ji programově upravit ani zjistit.

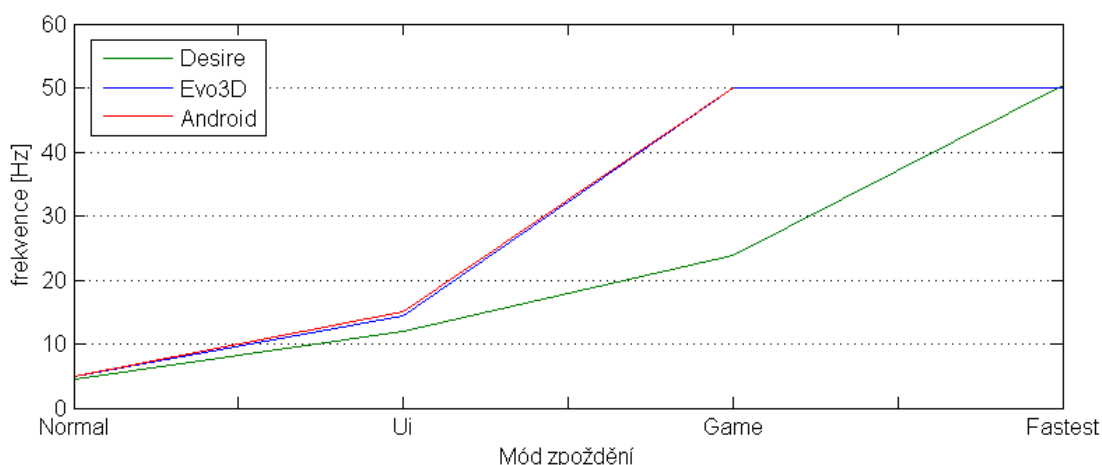
Měření probíhalo pomocí jedné z aplikací navržených pro tuto práci. Po dobu 10s je počítáno množství načtených vzorků. Z počtu vzorků a času je vypočtena frekvence. Tato aplikace je navržena tak, aby co nejméně zatěžovala telefon výpočty. Při běhu testu jsou všechny ostatní programy, které nejsou nutné pro běh systému Android ukončeny. Test byl spuštěn celkem desetkrát pro všechny módy zpoždění a oba telefony. Z těchto měření byl vypočten průměr, který je uveden v tabulce 2.4. Hodnotou Maximum Systému Android je hodnota vypočtená ze zpoždění jednotlivých módů, pokud by byl čas vzniklý šířkou pásma a zpožděním při výpočtu nulový (nebo téměř nulový), bude celý časový interval obsazen právě vloženým zpožděním systému Android.

Tabulka 2.4: Průměrné rychlosti odečítání

Název	Normal	Ui	Game	Fastest
Evo3D	5 Hz	14,29 Hz	50 Hz	49,98 Hz
Desire	4,42 Hz	12,044 Hz	23,88 Hz	50,46 Hz
Maximum systému Android	5 Hz	14,999 Hz	50 Hz	-

Z tabulky 2.4 tedy vidíme vliv rychlosti výpočetní jednotky telefonu na množství vzorků. Výpočetně rychlejší Evo3D kopíruje maximální rychlost danou systémem Android. Zpoždění dané šířkou pásma akcelerometru a rychlostí výpočetní jednotky je tedy téměř nulové. Při nulovém zpoždění módu fastest však množství vzorků dále nevzrůstá, jejich počet je přibližně stejný jako u módu game. Jestliže máme stejně vzorků, výpočetní náročnost bude stejná, v našem případě přibližně nulová. Můžeme tvrdit, že jsme narazili na maximální šířku pásma akcelerometru použitého v zařízení Evo3D. U zařízení Desire rozdíl mezi maximální a naměřenou rychlostí průběžně vzrůstá, se stoupajícím počtem vzorků roste výpočetní náročnost. Pomocí grafu (obr. 2.13), který popisuje data tabulky 2.4 můžeme porovnat průběhy obou telefonů. Jelikož v módu fastest Desire překonává Evo3D, můžeme potvrdit, že maximální šířka pásma jeho akcelerometru je větší než 50 Hz. Podle katalogu BMA150 a vývoje našeho grafu je šířka pásma tohoto akcelerometru od výroby nastavena na 100 Hz.

V praxi je tedy nejlepší používat mód fastest s nulovým vloženým zpožděním. V případě náročnější aplikace vzroste nárok na výpočetní výkon telefonu. Tento čas můžeme získat právě použitím módu fastest. Řízení vzdáleného zařízení v reálném čase však neklade velký nárok na zpracování dat akcelerometru. Postačí nám i telefon pomalejší. Pokud bychom chtěli hluboce analyzovat velké množství dat, musíme hledat zařízení s vysokým výpočetním výkonem a akcelerometrem s nastavenou vysokou šířkou pásma. Jak bylo již zmíněno, šířka pásma není běžně udávaným parametrem. Maximální rychlost musíme odměřit nebo zjistit z neoficiálních zdrojů.

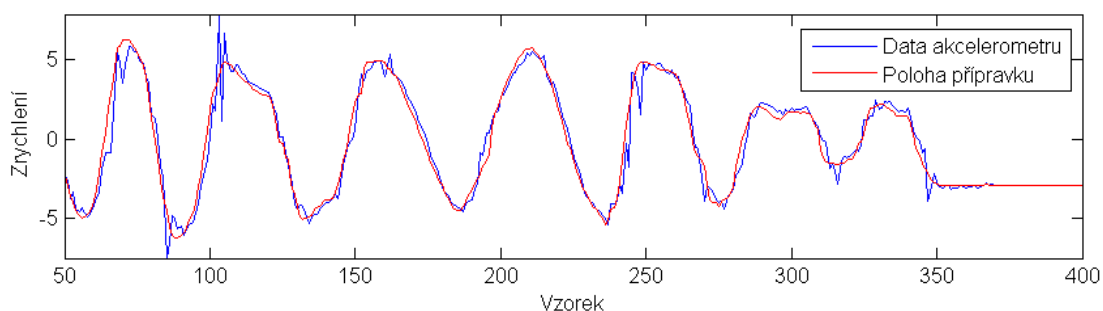


Obrázek 2.13: Porovnání rychlostí v jednotlivých módech

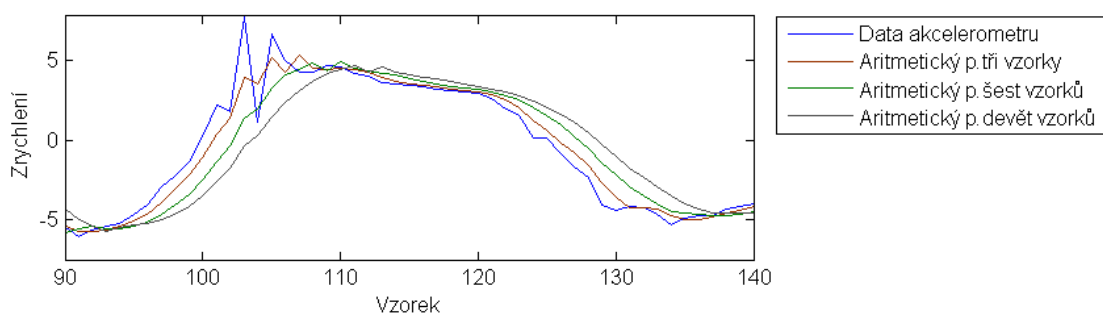
2.8 Úprava reálného signálu

Pokud budeme chtít vytvořit kvalitní, uživatelsky přívětivý produkt, musíme se vypořádat s problémy z předchozích kapitol. Zvolený způsob musí být kompromisem mezi výpočetním časem a výslednou funkčností. Jednoduchým a účinným řešením je použít správný typ průměrování vzorků. Základním typem je aritmetický průměr, který je nenáročný z hlediska výkonu. Druhým způsobem je získávat polohu pomocí středních hodnot, tedy pomocí mediánového vyhlazení. Tento způsob je však výpočetně náročnější v závislosti od metody zvolené pro řazení pole. Pro simulaci reálného pohybu je telefonem pohybováno rukou. Telefon je však stále upnutý v testovacím přípravku. Díky potenciometru v ose upnutého mobilního telefonu můžeme odečítat aktuální polohu v jedné ose pohybu. Z neupravených dat odečtených z akcelerometru osy x zařízení Evo3D (obr. 2.14) je na první pohled vidět šum a překmity při rychlé změně pohybu.

Pomocí aritmetického průměru můžeme jednoduše a výpočetně nenáročně vyhladit průběh signálu. Se stoupajícím počtem vzorků použitých pro průměr přirozeně vzrůstá posun od původního signálu. Na obrázku 2.15 s výsečí z kompletního grafu (obr. 2.14) můžeme tento posun pozorovat. Dále zjistíme, že průměr ze tří vzorků dostatečně nevyhladí náš signál. Pomocí devíti vzorků je vyhlazení dokonalé, daní je však větší posun od původních vzorků. Dobrým kompromisem je tedy volit průměr z šesti až devíti vzorků.



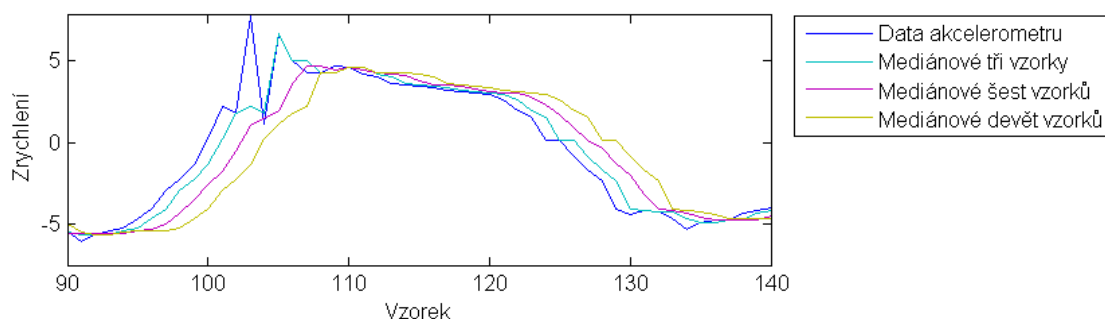
Obrázek 2.14: Porovnání reálného pohybu s daty akcelerometru



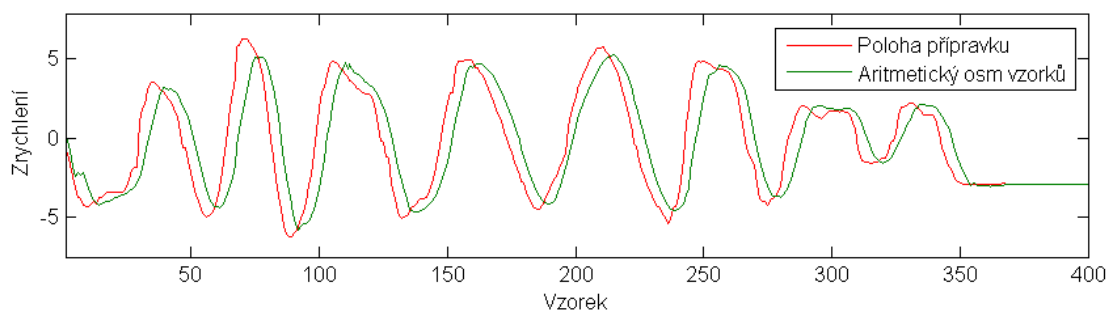
Obrázek 2.15: Porovnání aritmetického průměru podle množství vzorků

Mediánový průměr je dobrý, pokud chceme odstranit jednorázové překymty, které se v našem případě vyskytují při rychlém zastavení nebo rychlé změně pohybu. Na rozdíl od aritmetického průměru nebudou tyto výstřelové složky do průměru nijak započteny, a proto by neměl být záskmit při rychlém zastavení téměř nulový. Důležitou roli hraje výpočetní výkon nutný pro seřazení pole, ze kterého bude vybrána střední hodnota. Systém android v základu používá metodu `sort()`, která používá algoritmus Timesort s průměrnou složitostí $O(n \cdot \log n)$. Ačkoli použijeme maximálně devět vzorků, musíme vzít v potaz, že budeme pole třídit až padesátkrát za vteřinu. Pro mediánový průměr je vhodné použít lichý počet vzorků. Při použití sudého počtu je nutné vypočítat průměr mezi vzorky v polovině pole. Výsledný graf (obr. 2.16), porovnání vyhlazení pomocí tří, šesti a devíti vzorků, je počítačovou simulací. Není na něm tedy znatelný posun způsobený tříděním pole. Počet vzorků opět ovlivní posun signálu, podobně jako u aritmetického průměru. Tři vzorky jsou taktéž nedostačující.

Pro další potřeby této práce bude použito aritmetické vyhlazení signálu akcelerometru, které bylo díky své jednoduchosti a funkčnosti shledáno za nejlepší. Dobrým kompromisem je použít osm až deset vzorků pro vytvoření průměru. Na výsledném grafu (obr. 2.17) vidíme věrnost a posun vyhlazeného signálu oproti reálné pozici telefonu.



Obrázek 2.16: Porovnání mediánového vyhlazení dle množství vzorků



Obrázek 2.17: Porovnání reálného signálu se signálem upraveným

3 **Vhodné hardwarové prostředky**

Dnes je na trhu množství prostředků, jak propojit mobilní telefon s jiným zařízením. V podstatě postačuje jakýkoliv osmi bitový mikroprocesor doplněný o vhodné komunikační rozhraní. K dostání je velká škála mikroprocesorů, zejména od firmy Atmel a Microchip Technology. Komunikační modul, v našem případě bezdrátový, volíme v závislosti na technologiích podporovaných telefonem. Stejně jako v případě mikroprocesorů, je na trhu nespočet firem vyvíjejících bezdrátové komunikační moduly. Jak je již ve světě mobilních technologií zvykem, pro naše potřeby volíme ty, které jsou kompromisem mezi vysílacím výkonem a spotřebou elektrické energie. Lze však nalézt hotová řešení, která jdou jednoduše implementovat, upravit a legálně využívat ve výsledném produktu.

Dále si musíme ujasnit, jak budeme ovládat vzdálené zařízení. Pokud budeme ovládat stávající zařízení, které má své komunikační rozhraní například RS232 nebo X10, použijeme mikroprocesor s podporou těchto standardů. Díky této komunikaci nebudeme muset zasahovat do obvodového řešení původního zařízení, a jednoduše reprodukuje příkazy mezi telefonem a sběrnici stávajícího zařízení. Pokud budeme vytvářet vlastní obvodové řešení, ujasníme si co budeme chtít ovládat. Pokud bude v návrhu více integrovaných obvodů s komunikační sběrnici I²C, 1-Wire nebo například SPI, taktéž zvolíme vhodný řídicí procesor. Populární a jednoduše použitelný je digitální výstup s podporou pulzně šířkové modulace. PWM použijeme zejména pro kontrolu servomotorů, řízení jasu LED diod, řízení rychlosti motorů a další. Tento digitální výstup nalezneme ve většině mikroprocesorů, včetně nejlevnějších s několika vývody. Požadavky na procesor tedy vycházejí z problematiky návrhu elektronických obvodů a výroby plošných spojů.

3.1 **Umístění**

Celý systém bude obsahovat celkem tři části: mobilní telefon, vzdálené zařízení a námi vytvořený hardwarový systém. Umístění jednotlivých zařízení bude hrát nejdůležitější roli při výběru komunikačních protokolů. Pokud pomineme mobilní síť, které z hlediska odezvy nejsou vhodné pro plynulé řízení, ostatní bezdrátové technologie nabízené mobilním telefonem mají dosah jen několik metrů. Pokud však připojíme telefon i vzdálené zařízení pomocí bezdrátových technologií do sítě internet, dostaneme prakticky neomezený dosah. Můžeme tak spínat a ovládat různá zařízení v síti internet, která nemusí být řídicí osobou přímo viditelná. Důležitá je rychlá zpětná odezva.

V této práci však ovládáme vzdálené zařízení pomocí akcelerometru v mobilním telefonu. Již jsme zjistili, že pomocí námi testovaných zařízení získáme až 50 vzorků akcelerometru za sekundu. Využití akcelerometru je výhodné pro řízení pohonů, pro jednoduché spínání postačí tlačítka na displeji. Naše vzdálené zařízení bude tedy nejspíše v pohybu nebo bude něčím pohybovat. Pro plynulý pohyb s použitím maximálního možného počtu vzorků z akcelerometru budeme potřebovat dobrou zpětnou odezvu. Takovou odezvu lze zajistit pomocí videa. Přenos videa je však znatelně náročný a zvýší odezvu celého systému.

Pokud budeme ovládat zařízení v dohledu osoby, která ho pomocí akcelerometru v mobilním telefonu ovládá, tak je několik možností, jak umístit jednotlivé prvky, jejichž umístění značně změní nároky na obvodovou část. Zařízení umožňující komunikaci mezi telefonem a vzdáleným zařízením může ležet na straně telefonu, na straně ovládaného zařízení, na cestě mezi nimi nebo na obou stranách

3.1.1 Obvodová část na straně telefonu

Toto řešení použijeme zejména v případě, že máme již hotové zařízení, které chceme začít ovládat mobilním telefonem a nechceme zasahovat do jeho obvodů. Nutnou podmínkou je, že naše zařízení má již implementovaný nějaký způsob, kterým ho lze řídit na dálku. V tom případě může být naše obvodové řešení připojeno přímo k telefonu, například pomocí USB kabelu. Jako bezdrátová technologie bude použita ta, která je ve stávajícím vzdáleném zařízení již obsažena. Pokud je tato technologie standardizována, můžeme ji jednoduše implementovat v našem zařízení.

Výhodou tohoto řešení je, že nemusíme zasahovat do stávajícího zařízení, dále možnost napájení telefonu z našeho obvodu (nebo naopak) a přesnější kabelové spojení mezi telefonem a obvodem. Mezi nevýhody patří nemožnost změny bezdrátového protokolu, díky tomu není možné implementovat další funkce, které vzdálené zařízení nepodporuje. S dalším zařízením připevněným k telefonu vzroste váha, kterou musí uživatel držet v ruce. Čím více celé zařízení váží, tím je jemná manipulace, vyžadovaná při ovládání pomocí akcelerometru, těžší. Podobné řešení používají výrobci hraček, kdy je hračka ovládána dálkovým ovladačem. V tomto případě je použit přenos dat pomocí infračervených diod. K této hračce lze však dokoupit modul, který je připojen do audio konektoru telefonu. Díky němu můžeme ovládat hračku bez dalšího technického zásahu. Podobně můžeme použít infraport k přepínání televize. Toto řešení však není vhodné v kombinaci s akcelerometrem.

3.1.2 Obvodová část na straně ovládaného zařízení

Pokud používáme vlastní obvodové řešení celého vzdáleného zařízení, je pro nás tato konfigurace nejzajímavější. Použijeme stávající komunikační protokoly, které nabízí mobilní telefon. Můžeme tedy navrhnout jeden plošný spoj, který bude obsahovat veškerou funkcionalitu. Výhodou je vlastní řešení, které nás nijak nelimituje ve vývoji. Nevýhodou je nutná znalost elektroniky a tvorby plošných spojů.

Další možností je připojit náš obvod ke stávající drátové sběrnici původního zařízení. Většina zařízení pro automatizaci implementuje nějaký komunikační protokol, se kterým lze toto zařízení nastavovat a řídit (většinou na bázi RS232). Vytvoříme tedy převodník mezi bezdrátovou technologií a technologií drátovou. Tímto způsobem můžeme rozšířit funkcionalitu stávajících zařízení o bezdrátovou komunikaci. Dobrým příkladem může být integrovaný obvod ELM327 se sériovým bluetooth modulem. Pomocí tohoto obvodu můžeme vytvořit převodník, který lze připojit k automobilu. Pokud máme vhodný telefon s bluetooth rozhraním, můžeme s tímto převodníkem bezproblémově komunikovat a získávat hodnoty z řídících jednotek automobilu. Rozšíříme tedy funkčnost automobilu se standardním servisním konektorem o bezdrátovou technologii bluetooth.

3.1.3 Obvodová část uprostřed

V tomto případě jde o samostatný obvod, předávající informace z jedné bezdrátové technologie na druhou. Lze tedy přejít ze standardních protokolů na ty, které telefon přímo nepodporuje. Díky dvou bezdrátovým technologiím může všemi prvky volně pohybovat v prostoru. Tato možnost kombinuje výhody prvních dvou uspořádání. Nezasahujeme do původního zařízení, a máme stále ničím nezatížený telefon. Nevýhodou je oddělené napájení všech prvků. Pokud chceme náš obvod překládající signály napájet baterií, musíme vzít v potaz energetickou náročnost obou technologií.

3.1.4 Obvodová část na obou stranách

Tato možnost nám dává největší flexibilitu mezi všemi výše zmíněnými, a může tak volně kombinovat jejich výhody. Další výhodou je možnost implementace vlastního bezdrátového protokolu, který by nepodporoval telefon ani vzdálené zařízení. Díky této možnosti můžeme použít protokol se specifickým využitím. Technologie s pracovní frekvencí v řádu megahertz lépe prostupují překážkami, s řádem gigahertz tuto funkci ztrácí a nahrazují ji větší přenosovou rychlostí. Volba protokolu je tedy čistě na vývojáři, který určí pro danou aplikaci ten nejvhodnější. Nevýhodou tohoto řešení je dvojnásobná cena.

3.2 Komunikační protokoly

Dnešní telefony a ostatní mobilní zařízení integrují vesměs stejné komunikační protokoly. Nejpoužívanější jsou Bluetooth a WiFi. Možností je i starší několik let nepoužívané IrDA, které se možná znovu začne dostávat do telefonů (jako ovladač pro televizi). Díky dosahu a nutnosti přímé viditelnosti přijímače s vysílačem, je tato technologie pro náš účel nepoužitelná. V nejbližší době přijdou některé nové technologie, jako je například bezdrátové USB (Wireless USB). V současných zařízeních však stále nasazeny nejsou. Přesto můžeme s jistotou tvrdit, že WiFi a Bluetooth ještě nějakou dobu zůstanou dominantní.

3.2.1 WiFi

WiFi přichází nejprve pod standardem IEEE 802.11, který pracuje v bezlicenčním pásmu 2,4 GHz. Tato technologie je populární právě díky tomu, že za vysílání nemusíme platit licenční poplatky. Standard 802.11 nám poprvé umožnil připojit notebooky do lokální počítačové sítě. S rozvojem kapesních počítačů se přirozeně dostává i do nich. Postupem času dochází k rozšíření normy o 802.11b a 802.11g, kde dochází k navýšení přenosové rychlosti až na teoretických 54 Mbit/s. S vysokou oblibou a nízkým počtem kanálů (13 částečně překrývajících) brzy dochází k zarušení pásma 2,4 GHz. Alternativou je 802.11a pracující na frekvenci 5 GHz s několikanásobným počtem kanálů, které se nepřekrývají. Stejně jako na 2,4 GHz pracuje WiFi na 5 GHz v bezlicenčním pásmě [4].

Moderní telefony zvládají všechny zmíněné normy. Některé je rozšiřují o rychlejší 802.11n, které zvyšuje rychlost pomocí několika rozdílně polarizovaných antén. V mobilních telefonech používáme WiFi zejména pro připojení k internetu. Tato technologie není od počátku projektována s ohledem na spotřebu elektrické energie. Telefon při práci na baterii se zapnutým WiFi vydrží pouze několik hodin. Pokud použijeme WiFi pro komunikaci s naším zařízením, musíme počítat s relativně krátkou dobou používání a nemožností připojení k dalším sítím, například internetu.

Z pohledu obvodového řešení narazíme, jak na složitější hardwarovou, tak posléze i softwarovou implementaci. Stejně jako u mobilního telefonu bude spotřeba elektrického proudu vyšší, musíme proto počítat s vhodnou kapacitou baterie. Další problém je z hlediska finančního, kde jsou vysílací moduly pro WiFi znatelně dražší, než ty pro Bluetooth. Technologie WiFi je vhodná pro připojení mobilního telefonu i vzdáleného zařízení do počítačové sítě. Pokud budeme ovládat zařízení, které je v dohledu od řídící osoby, použijeme technologii Bluetooth.

3.2.2 Bluetooth

Bluetooth je primárně určen pro připojování periférií k notebooku nebo mobilnímu telefonu, kdy vytváříme takzvané privátní síť PAN (anglicky Personal Area Network). Bluetooth je od počátku projektován s ohledem na nízkou spotřebu elektrické energie. Zařízení napájené baterií vydrží mnohem déle se zapnutým Bluetooth, v porovnání s WiFi. Nyní existují čtyři verze a tři třídy Bluetooth. Verze vznikají postupem času, většinou zvyšují maximální rychlost a přidávají další funkcionalitu. Třída říká, jaký má modul odběr elektrické energie. S vyšším odběrem vzrůstá dosah. Jde tedy znovu o kompromis mezi úsporou energie a vysílacím výkonem. Bluetooth pracuje na frekvenci 2,4 GHz stejně jako WiFi, ale automaticky přepíná kanály, a tím zvyšuje odolnost proti rušení. Na rozdíl od WiFi pracuje i na vyšších vrstvách OSI modelu. Tím sice vzniká problém s kompatibilitou jednotlivých zařízení, ale obsáhne celý proces připojení a autentizace [5].

V praxi je modul Bluetooth modul jednoduše implementovatelný do obvodového řešení a jeho cena je oproti WiFi minimální. Díky sériové komunikaci je jednodušší i tvorba řídicí aplikace. Bluetooth nám tedy ušetří mnoho práce a sníží energetickou náročnost. Z uživatelského hlediska je také velice přívětivý, stačí pouze vyhledat a spárovat vzdálené zařízení. Problém může vzniknout v kompatibilitě modulu telefonu a vzdáleného zařízení. Zvláště systém Android verze 2.x má problémy s některými levnými moduly, tuto chybu lze v některých případech obejít. V dnešní době je tato starší verze systému Android na ústupu a v novějších je tento problém odstraněn.

3.3 Pohony a spínací prvky

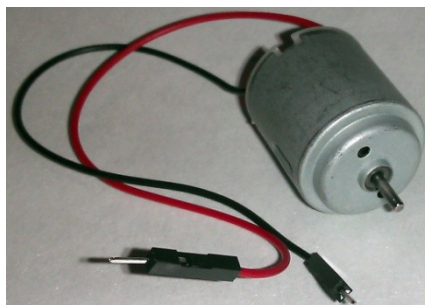
V této části bude popsáno několik pohonů, které nám umožní převést data akcelerometru na reálnou fyzikální veličinu. V našem případě půjde zejména o několik motorů na stejnosměrné napětí, díky nimž můžeme plynule reprezentovat pohyb telefonu. Jelikož je řídicí obvod také stejnosměrný, je pro nás použití těchto pohonů výhodnější. Pro řízení střídavých motorů použijeme spínací prvky, jako je například relé nebo stykač. Spínací prvky jsou vhodné pro jednorázové akce. Jejich sepnutí lze řídit tlačítkem na obrazovce telefonu, ale pro prezentaci dat akcelerometru moc vhodné nejsou.

Pokud se chceme vyhnout negativním vlivům, opotřebením a zničením motorů, vždy použijeme vhodnou úpravu signálu akcelerometru z kapitoly 2.8. Všechny pohony a spínací prvky byly otestovány a popsány autorem této práce.

3.3.1 Motor

Nejjednodušším pohonem je stejnosměrný motor. Stator je tvořen dvěma magnety s opačnou polaritou na jeho vnitřních stranách. Mezi magnety se nachází jednoduchý rotor se dvěma cívkami, kde díky kartáčkovému komutátoru je měněna jejich polarita. Díky změně polarity dojde ke změně magnetického pole rotoru mezi magnety statoru, a tím vznikne jeho pohyb. Stejnosměrný motor má pouze dva vývody (obr. 3.1). Otočením pólů vstupního napětí změním směr otáčení. Problémy těchto motorů leží právě v kartáčkovém komutátoru. Jelikož se kartáčky přímo dotýkají rotoru, přirozeně dochází k jejich opotřebení.

Pro řízení tohoto motoru pomocí mikroprocesoru potřebujeme doplnit návrh o řídicí obvod. Pokud chceme motorem otáčet pouze v jednom směru, postačí správně dimenzovaný tranzistor a jeden výstup mikroprocesoru. Pro možnost otáčení v obou směrech potřebujeme sestavit takzvaný H-můstek, který je tvořen celkem šesti tranzistory. Dva NPN a dva PNP výkonové tranzistory jsou zapojeny v schématu připomínajícím písmeno H (díky tomu název H-můstek). Zbylé dva slouží pro spínání výkonových tranzistorů z výstupu mikroprocesoru. Z toho vyplývá, že pro změnu otáčení musíme použít dva výstupy mikroprocesoru. Na trhu jsou některá hotová řešení, která nalezneme pod českým názvem „budič stejnosměrného motoru“. Podle toho, jaký výstup je v logické jedničce, takovým směrem se motor točí. Změna rychlosti se provádí pomocí PWM. Čím je menší rozestup mezi pulzy, tím je rychlost otáčení větší.



Obrázek 3.1: Kartáčkový stejnosměrný motor

3.3.2 Bezkartáčkový motor

Konstrukce je velice podobná motoru na střídavý proud. K dostání jsou dvě konstrukční varianty. Jedna s rotorem uvnitř a druhá s rotujícím pláštěm. Výhodou je nepřítomnost komutátoru. V motoru není žádné drátové spojení mezi rotorem a statorem. Z bezkartáčkového motoru nevedou dva vývody jako z klasického stejnosměrného, ale vývody tři (obr. 3.2). Tyto tři vývody jsou fáze, podobně jako u motoru na střídavý proud. Nutností je tedy řídicí obvod, který promění stejnosměrné napětí na jednotlivé fáze. Tento obvod je nazýván jako ESC (anglicky electronic speed control). Jeho cena je přibližně totožná s cenou motoru. S větším,

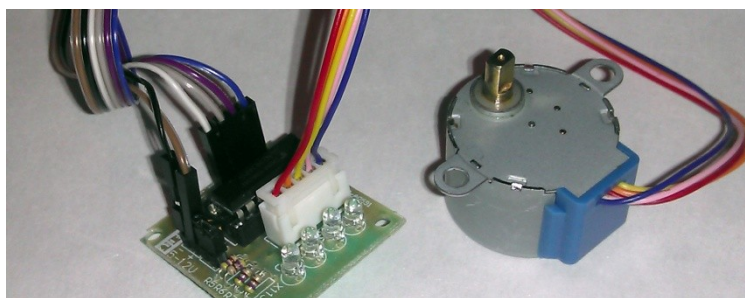
výkonnějším motorem je nutné i výkonnější ESC. Připojení k mikroprocesoru probíhá pomocí jeho PWM výstupu, kde jako u kartáčkového motoru řídíme rychlost šířkou pulzu. K připojení slouží tří pinový konektor, kde je se signálem veden zemnicí a napájení vodič (obr. 3.2). Poslední dva vývody ESC jsou při připojení k samostatnému zdroji energie. Přes svojí vyšší cenu dokáží tyto motory vyvinout více otáček a jsou proti klasickým kartáčkovým motorům méně náročné na údržbu. Jejich použití je zejména v počítačích, kde je nalezneme v mechanikách a pevných discích. Při změnách rychlosti a brzdění nedochází ke generování zpětného napětí, které ovlivňuje řídicí elektroniku. V jemné mechanice nalezneme tyto motory doplněné o sledovací kontakt, díky kterému máme zpětnou odezvu udávající rychlost motoru.



Obrázek 3.2: Bezkartáčkový stejnosměrný motor a jeho řídicí obvod

3.3.3 Krokový motor

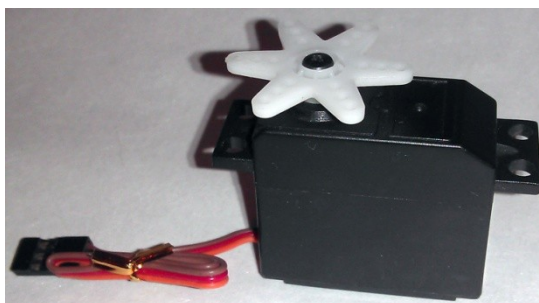
Krokový motor je ze všech pohonů nejpřesnější, proto se využívá v CNC frézách a dalších zařízeních s precizním pohybem. Limitem je jeho rychlost, která zdaleka nedosahuje rychlosti motorů ostatních. Samotný motor má většinou pět až osm vývodů v závislosti na jeho vnitřním zapojení (unipolární nebo bipolární). K jeho řízení je tedy nutný řídicí obvod (obr. 3.3). Pro jednodušší motory poslouží pole Darlingtonových tranzistorů, díky nim jsou v motoru vytvářeny pólové dvojice. Při jedné změně se motor otočí přesně o jeden krok. Díky tomuto krokování přesně známe polohu hřídele. Pro řízení krokového motoru potřebujeme N-1 digitálních vývodů mikroprocesoru, kde je N počet vývodů motoru (poslední je společná zem).



Obrázek 3.3: Základní krokový motor a jeho řídicí obvod

3.3.4 Servomotor

Servomotor je pohon s omezeným pohybem hřídele, většinou 0° až 180° . Uvnitř se nachází klasický kartáčkový motor, převody, potenciometr a řídicí obvod. Převody jsou situovány do pomala. To sice sníží rychlost, ale zvýší jeho sílu. Potenciometr je připojen přímo na hřídel vystupující z krytu. Díky němu řídicí obvod pozná přesnou polohu hřídele. Protože je řídicí obvod uvnitř, můžeme servomotor k mikroprocesoru připojit pouze pomocí jednoho třípinového konektoru (obr. 3.4). Tento konektor kromě signálu obsahuje i zem a napájení. Polohu nastavujeme taktéž pomocí signálů PWM. Šířka pulzů signálu však již neznamena rychlost, ale přesnou polohu hřídele. Průmyslové servomotory mají lehce jinou vnitřní strukturu. Vždy se zde nachází motor, převody a systém sledující polohu hřídele.



Obrázek 3.4: Servomotor

3.3.5 Relé

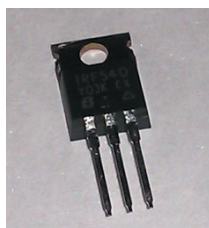
Relé je starý koncept spínacího prvku, který obsahuje řídicí vinutí a spínací, nebo rozpínací kontakty. Pokud přivedeme napětí na řídicí cívku, vzniklé magnetické pole posune kovovou kotvu, která sepne kontakty. Jedná se tedy o mechanickou součástku, která je ovšem náchylná na časté a zejména rychlé spínání. Přes vstupní kontakty je většinou doporučeno zapojit ochranou diodu, která zamezí zpětnému rázu při rozepnutí kontaktů (obr. 3.5). Díky relé můžeme malým vstupním napětím spínat velká napětí na spínacích kontaktech. Cívka a kontakty bývají většinou galvanicky odděleny. Dnes můžeme narazit na takzvané polovodičové relé, které je tvořeno tranzistory a neobsahuje pohyblivé části.



Obrázek 3.5: Relé s ochranou diodou

3.3.6 Unipolární tranzistor

Základní spínací součástkou v elektronice je tranzistor. Nejčastěji použijeme výkonový tranzistor MOSFET (z anglického metal–oxide–semiconductor field-effect transistor, obr. 3.6). Nastavením napětí na řídícím pinu (G - gate) pomocí mikroprocesoru docílíme průchodnosti proudu mezi piny ostatními (D - drain, S - source). Díky správně zvolenému tranzistoru MOSFET můžeme mikroprocesorem spínat napětí až 600V (v případě křemíkového tranzistoru). Na rozdíl od relé, které díky cívce nereaguje na PWM, můžeme tuto modulaci při tranzistoru MOSFET použít. Šířka pulzu je pak přenesena na řízené napětí.



Obrázek 3.6: Tranzistor MOSFET

3.4 Open Source Hardware

Open Source Hardware (OSHW) je licence umožňující volně šířit návrhy elektronických zařízení. OSHW vychází z Open Source Softwaru (OSS). Zde se však jedná o fyzická zařízení, licence je proto lehce upravena. Hardware pod touto licencí musí být tak veřejně přístupný, aby ho mohl každý studovat, upravovat, modifikovat a prodávat [6]. V praxi je ke každému výrobku veřejně dostupná dokumentace, schéma a návrh plošného spoje. Samotný uživatel může zakoupené zařízení podrobně studovat a upravovat. Tímto způsobem lze jednoduše a levně vytvořit prototyp. Důležitá je inovace, díky níž zůstávají v čele firmy, které se snaží svoje zařízení posunout dále. Pozadu zůstanou ti, kteří pouze kopírují předchozí výrobky. Největší roli v tomto odvětví hraje komunita lidí inspirující se určitým OSHW zařízením, kteří jej dále upraví dle svých potřeb. Tím vzniká mnoho variací původního návrhu. Pokud chceme námi upravené zařízení publikovat, musíme ho znovu vydat pod touto licencí, ale pod jiným jménem. Díky tomu dochází k opětovnému obohacování zainteresované komunity. Jestliže funguje produkt pod OSHW správně, brzy přitáhne masu lidí, která se mu začne aktivně věnovat. Noví uživatelé mají k dispozici silnou podporu v podobě rad, návodů a diskusních fór tvořených komunitou.

V našem případě můžeme využít několik OSHW projektů, které nám umožní bezdrátovou komunikaci s mobilním telefonem pomocí osmi bitového procesoru. Díky nim můžeme rychle a levně sestavit prototyp zařízení ovládaného pomocí akcelerometru.

3.4.1 Arduino

Arduino je exemplárním příkladem dobrého OSHW. V jeho čele je inovativní Italská firma Smart Projects podporovaná obrovskou komunitou lidí ze všech zemí světa. V budoucnu se již několik firem pokoušelo vytvořit jednoduchý způsob, jak programovat osmi bitové mikroprocesory. Arduino však poprvé použilo OSHW licenci a stalo se, díky tímto způsobem získané komunitě, nejpoužívanější platformou, jak pro laiky, tak i pro odborníky, kteří potřebují rychle vytvořit prototyp svého produktu. V roce 2010 došlo k prudkému nárůstu popularity, kdy bylo za tento rok prodáno 200 000 kusů této platformy. Arduino používá mikroprocesory ATmega firmy Atmel. U klasického mikroprocesoru ATmega musíme k programování použít speciální programátor a program. Díky přeinstalovanému zavaděči na desce Arduino programujeme mikroprocesor pomocí sériové linky (většinou virtuální přes USB) přímo z vývojového prostředí. Tvorba kódu je prováděna pomocí vyššího programovacího jazyka, který je postaven na jazyce Wiring. Tím se značně sníží vstupní investice a zjednoduší se celý proces programování.

Arduino nabízí mnoho bezdrátových modulů pro tvorbu spojení s mobilním telefonem, od výkonných WiFi, přes Bluetooth, až po jednoduché rádiové moduly. Arduino definuje vlastní název pro rozšiřující připojené desky, říká se jim anglickým slovem *shield* (štít). Tyto *shieldy* jdou většinou kupit nad sebe, a tím rozšířit samotné Arduino o množství dalších prostředků. Samotné Arduino umožní kontrolu všech zařízení z kapitoly 3.3 a rozšíří ho o další komunikační sběrnice jako je I²C, UART a další. Pro řízení motorů slouží nejrozumnější motor shieldy s řídicím obvodem pro daný typ motoru. Díky knihovnám tvořeným komunitou, můžeme všechny typy komunikace jednoduše implementovat do programu. Nejjednodušším způsobem, jak spojit mobilní telefon s deskou Arduino, je použít jednoduchý sériový Bluetooth modul. Jeho vysílací a přijímací piny přímo spojíme se sériovou sběrníci desky Arduino. Telefon spárujeme s Bluetooth modulem. Skriptovací vrstva pro android (SL4A) umožní tvorbu jednoduchých skriptů pomocí většiny známých skriptovacích jazyků (např. JavaScript, Python, Tcl a Ruby). Tímto skriptem můžeme jednoduše přistupovat k Bluetooth modulu stejně, jako k sériovému portu. Pod názvem Arduino BT můžeme koupit desku přímo s Bluetooth modulem, kterou vyrábí samotná firma Smart Projects. Tato deska vznikla jako reakce na mnohé uživatele, kteří používali k programování pouze Bluetooth modul.

Společnost Google v roce 2011 vydává svůj Accessory Development Kit (ADK), který přímo staví na platformě Arduino Mega 2560 (obr. 3.7) [7]. Google tak nabízí nejen firmám, ale i všem lidem možnost, jak jednoduše vytvořit prototyp hardwarového zařízení, připojeného k telefonu s operačním systémem Android. Komunikace probíhá pomocí protokolu Android

Open Accessory, který je detailně dokumentován samotným Google. Nevýhodou je nutnost drátového (USB) spojení mezi ADK a mobilním telefonem, proto nemůžeme ADK použít na straně vzdáleného zařízení.



Obrázek 3.7: Arduino ADK R3 přední a zadní strana [7]

3.4.2 Android IOIO

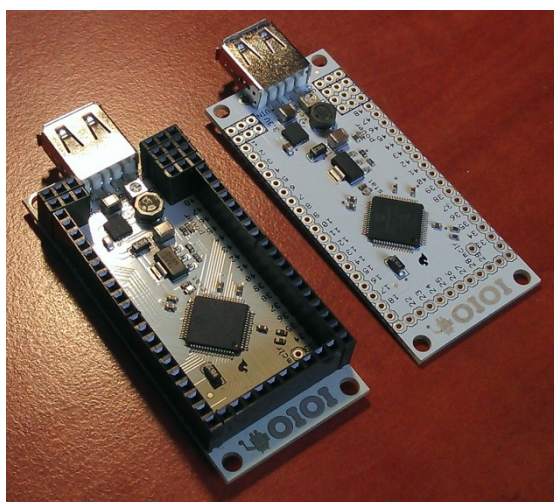
Android IOIO (dále jen IOIO) je speciálně zkonstruovaná deska pro komunikaci s operačním systémem Android. Název vznikl jako slovní hříčka prvních písmen anglického vstup/výstup (Input/Output) a hračky jojo. Konektivita je zprostředkována pomocí USB kabelu nebo Bluetooth modulu připojeného k desce IOIO. Tím pádem můžeme IOIO použít jak na straně vzdáleného zařízení, tak na straně mobilního telefonu. Samotný produkt vytvořil Izraelský autor Ytai Ben-Tsvi. Svůj nápad realizoval s kanadskou firmou SparkFun Electronics, která ho brzy rozšířila do celého světa. Tomuto projektu se díky OSHW dostává velké pozornosti a stává se nejjednodušší a nejlepší platformou pro tvorbu periférií pro Android. IOIO používá mikroprocesory PIC firmy Microchip Technology s nainstalovaným zavaděčem. Díky zavaděči nemusíme desku nijak programovat a vystačíme si se znalostmi tvorby aplikací pro Android. Celý kód je tedy tvořen v aplikaci pro mobilní telefon, kde stačí pouze importovat knihovny dodané k desce IOIO a použít třídu IOIOThread.

```
class IOIOThread extends AbstractIOIOActivity.IOIOThread {  
    @Override  
    protected void setup() throws ConnectionLostException {  
    }  
    @Override  
    protected void loop() throws ConnectionLostException {  
    }  
}
```

Třída IOIOThread běží pokud je IOIO spojeno s mobilním telefonem. Pokud ne, vrací standardní výjimku `ConnectionLostException`. Poté metoda `setup()` asociuje proměnné se vstupy a výstupy desky IOIO. Pokud dojde k rozpadu a vytvoření nového spojení, metoda `setup()` je volána znovu. Samotné vykonávání příkazů je prováděno v metodě `loop()`. Tato metoda je podobná běhu programu mikroprocesoru. Jedná se o nekonečný cyklus, který je vykováván až do doby rozpadu spojení. IOIO nám umožní komunikovat i pomocí protokolu Android Open Accessory firmy Google, ten však funguje pouze při připojení přes USB podobně jako ADK.

Jelikož v samotném mikroprocesoru neběží žádná námi vytvořená programová logika, ale pouze interpreter příkazů, tak v případě rozpadu spojení ztratíme kontrolu nad vzdáleným zařízením. V praxi dochází k blokaci výstupů mikroprocesoru, které zůstanou ve stejném stavu. V případě pohybujícího se vzdáleného zařízení toto zařízení nekontrolovaně odjíždí. Řešením je hardwarový hlídací obvod, který je v případě desky IOIO obtížně řešitelný.

Podobně jako u desek Arduino můžeme použít porty digitální, analogové, PWM, UART, I²C a další. Pro tuto práci byly otestovány dvě desky verze 1.6 (obr. 3.8) se zavaděčem verze 3.03 a knihovnami verze 3.23. V průběhu tvorby práce byla vydána nová verze 2.0 s názvem IOIO-OTG, která rozšiřuje původní verzi právě o USB On-The-Go specifikaci. Pokud je IOIO připojeno k zařízení se systémem Android, bude IOIO-OTG působit jako hostitel a bude dodávat nabíjecí proud do přístroje. Pokud ho připojíme k počítači s OS Windows nebo Linux, bude se IOIO-OTG prezentovat jako virtuální sériový port. Jak verze 1.6 tak 2.0 zachovávají poměrně malou velikost 77 x 32 cm, dají se tak jednoduše implementovat do stávajících zařízení.



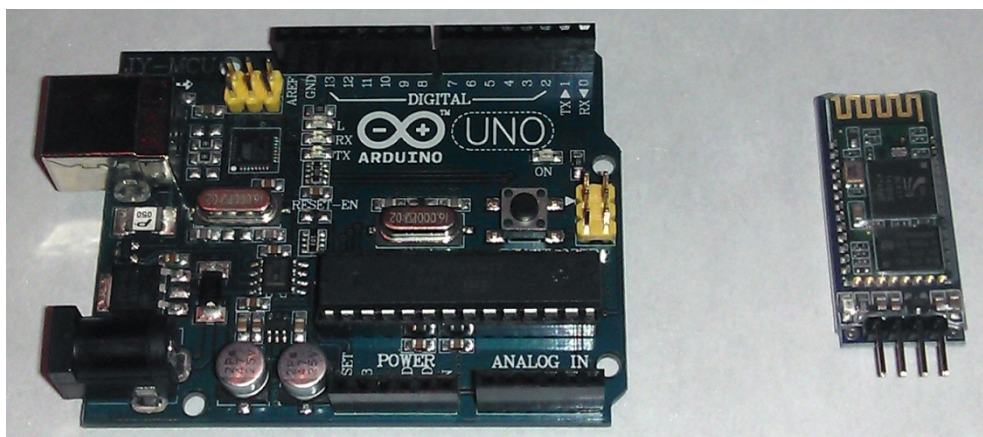
Obrázek 3.8: Android IOIO verze 1.8

3.4.3 Arduino klon a další

Pod pojmem Arduino klon se skrývá množství platforem vycházejících z projektu Arduino. Slovem klon jsou označovány jednak odnože tvořené pod licencí OSHW, tak i (většinou čínské) kopie původních desek, které jsou vydávány pod stejným jménem a tím porušují ujednání OSHW.

Jeden z projektů umožňujících bezdrátovou komunikace je Moteino, které navrhl Felix Rusu. Jedná se o extrémně malou variantu desky vycházející z projektu Arduino, kterou lze osadit bezdrátovým modulem. Celý obvod je velký přibližně jako paměťová karta SD, tím ho lze jednoduše integrovat do nejrůznějších zařízení. Dalším vhodným projektem je MBoard, který na jedné desce integruje řídicí obvod kartáčkových motorů a patici pro XBee modu. XBee je rodina komunikačních modulů s různou přenosovou technologií, ale stejnou patičkou a stejným programovým přístupem. XBee nabízí WiFi, Bluetooth a další moduly, které sice nejsou přímo kompatibilní s totožnými protokoly v mobilním telefonu, lze je však zapojovat do mesh a senzorových sítí, které lze řídit z jednoho přístupového bodu, který bude do sítě interpretovat data akcelerometru. Projektů vycházejících z Arduina je opravdu hodně. Na nás je najít ten správný, nebo podobný a vhodně ho upravit. Díky OSHW licenci, která všechny tyto projekty zastřešuje, nejsou kladeny žádné limity pro fantazii vývojáře.

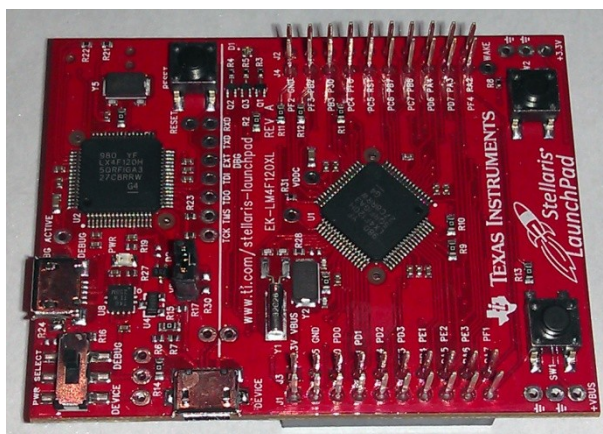
Dalším případem jsou přesné kopie desek Arduino vydávaných jinou společností. Tímto chováním porušují licenci OSHW a dopouští se ho zejména Čínští výrobci. Jejich desky jsou přesnou kopií těch originálních. Rozdíl je však v kvalitě použitých součástek a způsobu výroby. Pro tento projekt byl otestován čínský klon desky Arduino UNO firmy JY-MCU (obr. 3.9 vlevo) se sériovým Bluetooth modulem BT_BOARD V1.02 stejné firmy (obr. 3.9 vpravo). Komunikace probíhala pomocí skriptovací vrstvy SL4A zmíněné v kapitole 3.4.1.



Obrázek 3.9: Klon Arduino UNO vlevo, Bluetooth modul vpravo

Pokud nám nevystačí jednoduchý osmi bitový procesor, můžeme použít výkonnější procesorovou platformu ARM. ARM je zkratkou Advanced RISC Machine, používající 32bitovou architekturu a je nedělitelnou součástí většiny všech výkonných mobilních zařízení. Díky této architektuře může na zařízení běžet samostatný operační systém, jako je například Linux nebo Android. Dnes je velice populární platforma Raspberry-pi firmy Raspberry Pi Foundation. Tato platforma vznikla z potřeby vyvinout jednoduchý a levný počítač pro výuku informatiky, což se jí podařilo. Raspberry-pi nepoužívá přímo OSHW, ale Creative Commons - ShareAlike. Tato licence umožní původnímu tvůrci zakázat vyrábět klon jeho zařízení, pokud nový výrobce nedodrží ostatní podmínky této licence. Na Raspberry-pi běží upravená verze systému Linux. Lze připojit monitor a periferie běžné pro osobní počítač. Výhodou jsou digitální porty, které nabízí propojení s námi vytvořeným hardware podobně jako u platformy Arduino. Pomocí USB portů rozšíříme Raspberry-pi o Bluetooth nebo WiFi rozhraní. Můžeme tedy použít výkonnou platformu, která umožní provádět výpočty mnohem rychleji než osmi bitový mikroprocesor. Procesor ARM implementovalo samotné Arduino v platformě zvané Arduino Due.

V komunitě tvořené okolo OSHW vidí sílu mnoho velkých firem. Jednou z nich je firma Texas Instruments, která se snaží rozšířit svůj vývojový kit Stellaris LaunchPad LM4F120 (obr. 3.10). Ten je osazen 32bitovým procesorem ARM Cortex M4 s taktem 80 MHz. Texas Instruments slibuje jednoduché použití jeho vývojového kitu, snadnost použití však zdaleka nedosahuje jednoduchosti a rychlosti použití ostatních systémů. Pro tuto práci byl jeden Stellaris LaunchPad pořízen. S ohledem na dostačující výkon osmi bitových mikroprocesorů však nebyl použit. Přesto došlo k instalaci potřebného prostředí a ověření základní funkčnosti. Výkon osazeného procesoru mnohonásobně přesahuje jakýkoli osmi bitový mikroprocesor, Pro náročnější aplikace zůstává dobrou alternativou k ostatním ARM platformám.



Obrázek 3.10: Vývojový kit Stellaris LaunchPad LM4F120

4 Vlastní řešení

V této kapitole bude popsána samotná realizace vlastního zařízení, kde bude využito všech znalostí získaných z předchozích kapitol. Pro tuto práci je vytvořeno hned několik zařízení, které demonstrují většinu získaných poznatků z částí teoretických. Všechna řešení budou spadat do množiny rádiově řízených modelů. Na první pohled se může zdát, že se jedná o okrajovou záležitost a o hračky. Samotný trh s dálkově řízenými modely je však velký nejen u nás, ale i ve světě. Většina modelů zasahuje do cenové oblasti pět až deset tisíc korun. Pokud bude naše zařízení použitelné v praxi nebo se i podaří ušetřit za drahý rádiový ovladač (výměnou za mobilní telefon), může mít naše řešení velký úspěch. Na modelech lze navíc dobře demonstrovat praktiky, které lze použít při řízení velkých pohonů a prvků, které již modely nejsou a slouží lidem k práci a každodennímu životu.

4.1 Obvodová řešení

Navržená řešení budou vycházet z licence Open Source Hardware, zejména z platformy Android IOIO popsané v kapitole 3.4.2. Android IOIO bylo vybráno jako nejlepší prostředek pro komunikaci s OS Android. Pro práci byly zakoupeny dvě tyto desky (obr. 3.8) s firmware ve verzi 3.02. Podpora bluetooth modulů je již od verze 3.0. Z doporučení vývojářů byla nahrána v tu dobu aktuální verze 3.11, která vylepšila zejména zmíněnou bluetooth konektivitu. Výhodou změny firmware na deskách Android IOIO je, že není potřeba další programátor. Pro přehrání použijeme druhou desku připojenou k mobilnímu telefonu s aplikací IOIO Manager (autor Ytai Ben-Tsvi). Celý proces aktualizace firmware je popsán na stránkách repozitáře platformy Android IOIO uživatele ytai, kde nalezneme i celkovou dokumentaci k projektu [8].

4.1.1 Použití stávajícího ovladače

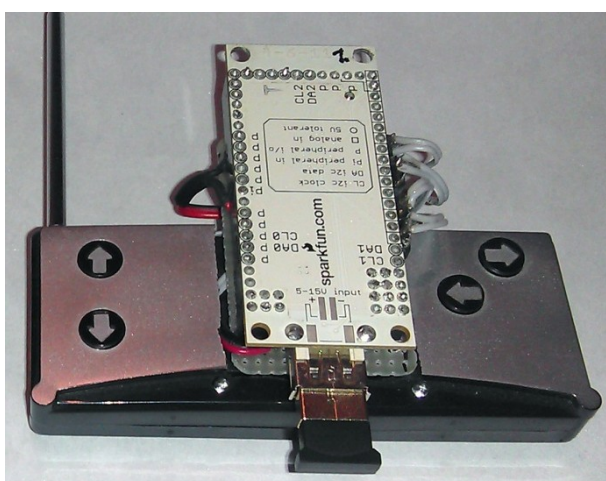
Tato část navazuje na teorii z kapitoly 3.1.3, kdy obvodové řešení leží uprostřed komunikační cesty. V našem případě máme jednoduchý a levný model dálkově řízeného automobilu pomocí bezdrátového ovladače (obr. 4.1). Ovladač pracuje na frekvenci 35Mhz a je tzv. dvoukanálový. To znamená, že můžeme řídit pohyb modelu ve dvou směrech, pohon dopředu/dozadu a zatáčení doprava/doleva.

Pokud chceme řídit tento model pomocí mobilního telefonu, musíme vzít v potaz, že náš telefon nepodporuje komunikaci pomocí bezdrátové technologie na frekvenci 35Mhz. Tento konkrétní model je rozměrově tak malý, že je takřka nemožné rozšířit ho o mikroprocesor a bluetooth modul tak, aby se změna velikosti neprojevila na jeho zevnějšku. Můžeme však využít služby stávajícího ovladače, a ten vhodně rozšířit o bluetooth rozhraní.

Velikost ovladače nám umožní vhodně osadit vývojovou desku Android IOIO přímo na jeho tělo. Výsledkem bude ovladač, který funguje stejně jako dříve, ale slouží i jako převodník informací mezi mobilním telefonem a protokolem, který model používá. Výsledný ovladač nalezneme na obrázku 4.2. Ten je osazen jednak deskou Android IOIO, tak i dalšími obvody, jejichž schéma naleznete v příloze E. Hlavními částmi jsou optočleny 4N35, které ve schématu nalezneme pod označením OK1 až OK4. Tyto součástky slouží ke galvanickému oddělení obvodů Android IOIO a obvodů ovladače. Díky tranzistorům v součástce 4N35 můžeme spínat kontakty tlačítek (B1 až B4 ve schématu). Dále jsou použity rezistory R1 až R4 s hodnotou 200Ω , které slouží jako ochranný odpor pro diody optočlenů. Rezistory R5 až R8 s hodnotou 1000Ω slouží pro nastavení logické nuly, pokud není na výstupech mikroprocesoru (svorky IOIO_PINS_1-4 ve schématu) nastavena logická jednička. V praxi nejsou odpory R5 až R8 nutné, je však doporučeno tímto způsobem ošetřovat digitální vstupy a výstupy mikroprocesorů. Poslední částí je regulátor napětí. Android IOIO vyžaduje vstupní napětí alespoň 5V a ovladač maximálně 4V. Pokud chceme napájet výsledný produkt z jednoho zdroje, musíme vstupní napětí pro ovladač vhodně snížit. Použijeme tedy 9V baterii typu PP3, která bude umístěna v původním držáku pro baterie ovladače. Napětí baterie bude přivedeno na vstupní regulátor desky Android IOIO a na náš regulátor, který sníží napětí pro ovladač.



Obrázek 4.1: Původní model s ovladačem



Obrázek 4.2: Upravený ovladač

Výsledný produkt je kopií původního ovladače, rozšířeného o bluetooth rozhraní. Je tak dobrým popisem zařízení z teoretické kapitoly 3.1.3. Jelikož původní ovladač neumožňoval plynule řídit rychlost a natáčení kol, neumožní ho ani nový ovladač rozšířený o bluetooth. V našem případě nemůžeme náklonem telefonu plynule zrychlovat a zatáčet, můžeme pouze nastavit plnou rychlost a natočení kol po překročení námi zvoleného úhlu náklonu telefonem. Pokud chceme naše zařízení rozšířit o další funkce musíme upravit obvodovou část samotného modelu.

4.1.2 Řídící obvod na straně vzdáleného zařízení

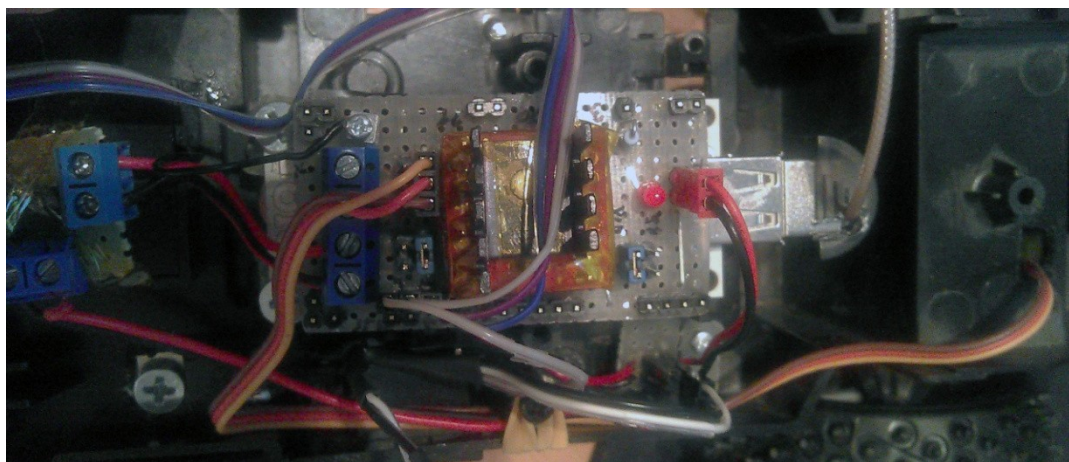
Toto řešení nabídne vývojáři velkou volnost co se týče tvorby obvodové části, zde je vše ponecháno pouze na jeho fantazii. Jediné omezení tvoří volba bezdrátové technologie, kterou musí mobilní telefon podporovat. K této části se vztahuje teoretická kapitola 3.1.2. Jako řídicí platforma bude použito též Android IOIO.

Pro tuto část byl upraven již větší model dálkově řízeného automobilu. Kromě motoru pro pohyb dopředu a dozadu byla kompletně odstraněna veškerá elektronika. Nepřesný systém zatáčení, který se skládal ze stejnosměrného motoru a pružiny, byl nahrazen přesnějším servomotorem. Původní kazetová 4,6 V baterie byla nahrazena čtyřmi bateriemi typu AA. Nové obvodové řešení je popsáno schématem v příloze F. Hlavní součástí obvodu je budič stejnosměrných motorů TB6612FNG (U1 ve schématu), jeho zapojení částečně vychází z knihy „Making Android Accessories with IOIO“ [9]. Toto zapojení však nebylo dokonalé, chyběly ochranné diody (D2 až D9 ve schématu), které zabrání vstupu napěťových špiček do obvodu. Tyto špičky vznikají jednak při brždění, tak i při samotném chodu kartáčkového motoru, kvůli občasným zkratům mezi kartáčky. Bez těchto diod dochází k výpadkům činnosti mikroprocesoru, tedy i výpadkům komunikace. Samotný budič zvládne ovládat dva stejnosměrné motory o maximálním odběru až 2 A. V našem zařízení je použit pouze jeden motor. Na druhý motor je však obvod připraven. Svorky motorů jsou ve schématu označeny jako MA a MB. K řízení jednoho motoru slouží vstupy PWM, AIN1 a AIN2 budiče TB6612FNG. Vstup PWM určuje rychlost dle šířky pulzu, kterou nastaví mikroprocesor. Směr otáčení je volen kombinací logických jedniček a nul na pinech AIN.

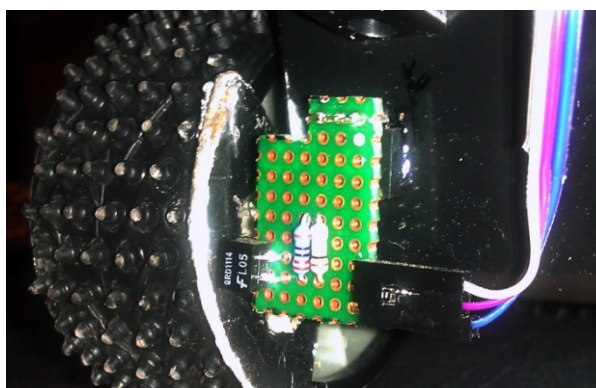
Svorkovnice SL2, SERVO1 a SERVO2 slouží pro připojení periférií. Ke svorkovnici SERVO1 je připojen servomotor, který zatáčí předními koly modelu. SL2 slouží pro připojení optického senzoru a tranzistoru MOSFET, jejich funkce bude popsána níže. Svorka SERVO2 je neobsazena a lze k ní připojit libovolný servomotor. Dále je osazena stavová dioda D1 s předřadným odporem R1, která slouží k indikaci libovolného stavu. Veřejněji indikuje stav

spojení s telefonem. Periferie jsou připojeny vždy k příslušným portům desky Android IOIO dle schématu. Piny jsou označeny jako svorky začínající názvem IOIO_PINS_. Výsledný obvod zapojený na univerzálním plošném spoji je zobrazen na obrázku 4.3.

Řídící obvod je rozšířen o dva doplňkové obvody, které jsou zapojeny do SL2 pomocí svorky SL1 na jejich straně. Prvním je tranzistor IRF530, který slouží pro spínání dekorativního osvětlení modelu znázorněného diodou LED1 ve schématu. Jednoduchou LED diodu lze spínat slabším tranzistorem nebo samotným výstupem mikroprocesoru. Model je však osazen dvěma LED a dvěma laserovými diodami, zapojenými paralelně. Ze samotného pinu mikroprocesoru by již tak velký proud nemohl být odebírán, mohlo by dojít k jeho poškození. Posledním rozšiřujícím obvodem je optický odrazový spínač QRD1114. Rezistor R3 slouží jako předřadný odpor pro infračervenou diodu optického spínače. R2 je použit pro nastavení logické jedničky v případě, že tranzistorem není zachyceno infračervené světlo z diody. Tento malý obvod je umístěn v blízkosti zadního kola, na kterém je připevněna odrazová plocha (obr. 4.4). Díky tomuto nastavení lze zachytit jednu otočku kola mikroprocesorem. Podle počtu otáček vypočteme rychlost našeho modelu.



Obrázek 4.3 Řídící obvod modelu



Obrázek 4.4: Optický spínač sledující otáčení kola

O přenos dat se stará jednoduchý sériový USB bluetooth modul. Jedná se o nejlevnější modul, který je dnes k dostání pod názvem „Super Mini Bluetooth 2.0 Adapter Dongle“. Dosah této konfigurace je pouze přibližně 10 m a proto byla přepracována anténí část modulu. Spoj integrované antény byl přerušen, na jeho místo byl připájen prodlužovací kabel, takzvaný *pigtail*, s konektorem SMA. Tento konektor byl vyveden ven z modelové karoserie a upevněn na zadní část modelu. Díky němu lze připojit libovolnou 2,4 GHz anténu, a tím získat větší dosah. V našem případě byla osazena jednoduchá WiFi všesměrová prutová anténa s přibližným ziskem 5dB. Díky stejné pracovní frekvenci protokolů bluetooth a WiFi, tedy 2,4 GHz, můžeme jejich antény libovolně zaměnit. Za pomoci této jednoduché všesměrové antény vzrostl dosah na hranici 30-ti metrů.

4.1.3 Univerzální modelářské řešení

Tento koncept vychází z myšlenky nahradit univerzální modelářský rádiový vysílač a přijímač mobilním telefonem. Toto řešení má za cíl snížit cenu dálkově řízeného kompletu, jelikož cena samotného vysílače tvoří až třetinu jeho ceny. Cena radiového vysílače závisí zejména na počtu jeho kanálů. Počtem kanálů je myšleno počet výstupů s určitou pulzní modulací, které můžeme k řízení modelu využít. Pokud vyměníme rádiový vysílač za mobilní telefon a přijímač za desku Android IOIO, budeme mít k dispozici až deset kanálů s PWM modulací. V tabulce 4.1 můžeme vidět jednoduchou kalkulaci cen stávajících rádiových vysílačů (cena i s přijímačem), v porovnání s naším řešením, které využívá desku Android IOIO. Uvedené ceny jsou získány na základě internetového vyhledávače cen Heureka dne 10.4.2013. V tabulce byly použity vždy nejnižší nalezené ceny. V ceně konfigurace s Android IOIO je započtena cena bluetooth modulu a dalších elektronických součástek v přibližné hodnotě 250Kč. Cena samotného mobilního telefonu s OS Android započtena není, protože předpokládáme, že uživatel volící toto řešení již mobilní telefon vlastní.

Tabulka 4.1: Cenová kalkulace vysílačů

Rádiový komplet	Počet kanálů	Cena
Turnigy 9x	8	2 269 Kč
Futaba 10CG	10	14 088 Kč
Android IOIO	10	1 217 Kč

V praxi postačují dva kanály pro řízení modelů automobilů a čtyři kanály pro řízení letadel a vrtulníků. Uživatelé většinou využijí další dva kanály pro ovládání nejrůznějšího příslušenství. Vidíme, že zmíněné komplety bez problému pokryjí potřeby většiny uživatelů. Turnigy 9x je populární bezdrátová souprava, která v dnešní době nabízí nejlepší poměr ceny

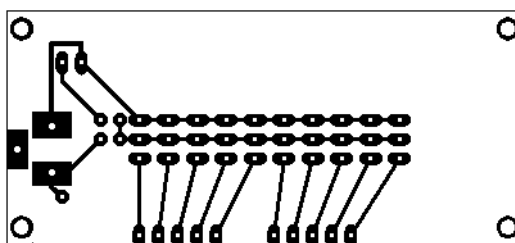
a počtu kanálů. Zařízení s deskou Android IOIO však nabídne dva kanály navíc za téměř poloviční cenu. Na obrázku 4.5 je porovnávána souprava Turnigy 9X a naše řešení. Prototyp byl vytvořen na univerzálním plošném spoji jako nadstavba na Android IOIO. Dále bylo vytvořeno schéma (Příloha G) v programu Eagle firmy CadSoft (verze 6.2.0 pro Windows, Light Edition), z tohoto schématu byl dále vytvořen návrh plošného spoje (obr. 4.6). Na vrchu schématu je znázorněn standardní 2,5 mm napájecí konektor. Ten však nemusí být osazen. Většina řídicích obvodů pro motory používaných v rádiově řízených modelech nabízí možnost napájení řídicích jednotek. To znamená, že můžeme napájet desku Android IOIO pomocí některých ze svorek SV1 až SV10, které slouží právě pro připojení řízených prvků. Tyto svorky mají dané rozložení pinů. První je signál, druhý napájení a třetí je zem. Toto rozložení je použito u všech prodáváných přijímačů, můžeme tak přímo připojit nejrůznější pohony a servomotory. Svorky SV11 a SV12 znázorňují spojení s digitálními výstupy desky Android IOIO, které podporují PWM. Svorka VIN zajistí spojení se stejným pinem na řídicí desce. Pokud osadíme napájecí konektor, toto napětí bude přivedeno na napěťový regulátor naší desky. Svorka VIN/5V slouží pro volbu napájecího napětí pro svorky SV1 až SV10 pomocí dvou pinové propojky. Poslední svorkou je 5V/GND. Ta slouží jednak pro vyrovnání potenciálů pomocí propojení zemí, tak i pro propojení 5 V napěťových větví.



Obrázek 4.5: Souprava Turnigy 9x vlevo a souprava s Android IOIO vpravo

Řešení s Android IOIO je tedy velice jednoduché a levné. Bezdrátové soupravy však v některých směrech plně nenahradí. Prvním problémem je zejména dosah, který závisí jednak na osazeném bluetooth modulu na straně přijímače, tak i na modulu v mobilním telefonu. Na straně přijímače s Android IOIO můžeme použít libovolný modul s libovolnou anténou. Případně použít zesilovač, a tak efektivně prodloužit dosah. Na straně mobilního telefonu již tato možnost není. Dnešní telefony nenabízí připojení externí antény pro bluetooth, ani rozumnou výměnu celého bluetooth modulu. Jak již bylo zmíněno, největší zjištěný dosah byl 30 m, a to při použití malé antény na straně modelu. Porovnávané Turnigy 9x nabízí dosah až 1 km. V praxi bohatě dostačuje dosah 300 m, při kterých přestává být model viditelný okem. V mobilním telefonu nalezneme nejčastěji bluetooth modul *class 1*, a to bez větší antény. To dle specifikace bluetooth znamená, že získáme dosah přibližně 100 m. Mobilní telefon se tedy nemůže nikdy plně vyrovnat dosahu rádiových souprav.

Druhým problémem je celková stabilita a kvalita spojení. Ačkoli moderní rádiové soupravy pracují na frekvenci 2,4 GHz stejně jako bluetooth, jejich protokol je jiný. Protokoly používané v rádiových soupravách jsou primárně určeny k řízení modelů automobilů a letadel. Zejména u modelů letadel je kvalita spojení nejdůležitějším prvkem. Pokud dojde ke ztrátě spojení, přijdeme o model, který stojí řádově tisíce. Protokoly používané v rádiových soupravách rychle navazují rozpadlé spojení a snaží se být co nejvíce odolné proti rušení. Jeden z pilířů bluetooth je také odolnost proti rušení. Protokol však není k našemu účelu přímo určen. Díky jeho univerzálnosti je přenášeno mnohem více dat, nežli je k ovládání modelů potřeba. Při rozpadu spojení trvá nové navázání někdy i několik sekund. Takové zpoždění je při řízení modelů letadel nemyslitelné. Někdy je nutné i znovu spárovat obě zařízení. Výsledný obvod lze rozšířit o různé hlídací obvody. V případě modelů letadel však bude jakýkoli delší výpadek znamenat pád. Proto je toto řešení vhodné zejména pro řízení modelů automobilů.



Obrázek 4.6: Návrh plošného spoje univerzální nastavby pro Android IOIO.

Pro modely letadel lze využít Android IOIO v kombinaci se stávající rádiovou soupravou. Můžeme tak rozšířit funkce stávajícího ovladače o další kanály. Díky univerzálnosti desky Android IOIO můžeme osadit další senzory, jako jsou akcelerometry, gyroskopy, teplotní senzory, senzory tlaku, čidlo měření napětí baterie modelu nebo GPS modul. Výstup těchto senzorů sledujeme na obrazovce mobilního telefonu a řízení přenecháme na bezdrátové soupravě. Při výpadku spojení s telefonem můžeme model stále řídit. Výsledkem bude kvalitní telemetrický systém, který je u leteckých modelářů velice populární a jeho cena bývá také vysoká. Zde je vše znovu ponecháno na fantazii vývojáře, který může díky této platformě využít všechny prostředky z běžné elektroniky.

4.2 Řídící aplikace

Plnohodnotná aplikace byla vytvořena pro první dvě řešení. U univerzálního modelářského řešení je tvorba aplikace značně náročná a zasahuje mimo rámec této práce. Tato práce je zaměřena na ovládání vzdáleného zařízení pomocí akcelerometru v mobilním telefonu. Všechny techniky mohou být dobře demonstrovány na prvních dvou příkladech.

Obě aplikace jsou napsány pro OS Android a pro vývojovou platformu Android IOIO. Při vývoji bylo použito prostředí Eclipse (ver. 3.6.2), které je nativním vývojovým prostředím pro systém Android. Z důvodu testování i na starším zařízení Desire bylo použito Android SDK (software development kit) pro verzi Android 2.2 (API 8). Starší aplikace lze použít i na novějších verzích systému android, stejnou aplikaci lze tedy použít i na EVO3D s OS Android verze 4.0.3 (API 15). Dalším krokem je vývojový kit pro jazyk Java, tedy JDK (Java development kit) ve verzi 6. Dle dokumentace Android IOIO musí být použito právě JDK verze 6. Programování probíhá stejně jako u klasické aplikace pro OS Android. Jediné co potřebujeme je importovat knihovny IOIOLib a IOIOLibBT do našeho projektu, které zajistí komunikaci s deskou Android IOIO. Obě knihovny jsou volně ke stažení v repozitáři projektu Android IOIO [8]. Ovládání portů desky probíhá ve třídě IOIOThread zmíněné v kapitole 3.4.2. Důležité je v souboru AndroidManifest.xml povolit přístup aplikace k bluetooth rozhraní a ke komunikačním protokolům. To můžeme provést jak v grafickém editoru, tak pomocí kódu.

```
<uses-permission android:name="android.permission.BLUETOOTH"/>
<uses-permission android:name="android.permission.INTERNET"/>
```

Tuto část kódu je nutné zapsat ihned pod část uvádějící verzi použitého API začínající tagem `<uses-sdk . . .` přímo v kódu souboru AndroidManifest.xml.

Aplikace pro zařízení využívající stávající ovladač je velice jednoduchá. Aplikací je ovládána čtveřice digitálních výstupů, které spínají jednotlivá tlačítka ovladače. Výstupy jsou v programu zastoupeny proměnnými d1_ až d4_ a jsou spínány v případě, že jsou stlačena příslušná tlačítka, která zastupují proměnné b1_ až b4_. Pokud je zaškrtnuto pole s názvem „sensor mode“, které je reprezentováno proměnnou ch1_, pak aplikace použije data akcelerometru. Data jsou získávána kódem, který byl popsán v kapitole 2.2. Jelikož ovládáme pouze tlačítka, není nutné data akcelerometru dále upravovat a vyhlazovat. V programu je nastavena prahová hodnota pro každý směr naklonění telefonu. Pokud je hodnota větší, je aktivován příslušný výstup mikroprocesoru, a tím je sepnuto tlačítko ovladače.

Již komplexnější aplikace slouží k ovládání druhého zařízení z kapitoly 4.1.2. Tělo aplikace zůstává stejné, stejně tak i způsob získávání dat z akcelerometru. Hardware nám však umožní plynulé řízení rychlosti a plynulé natáčení předních kol, aplikace tedy musí této možnosti využít. Proto jsou data akcelerometru ukládána, do dvou polí datového typu float s názvem mSensorX a mSensorY, podle osy náklonu x a y. O ukládání hodnot se stará metoda addSensorValue(), která uloží hodnoty akcelerometru, do jejich polí, vždy při jejich změně. Pomocí této metody jsou data posunuta v polích tak, aby byla nahoře vždy nejnovější hodnota. Hodnoty z polí získáváme metodami MakeAverageX() a MakeAverageY(), které provedou aritmetický průměr a vrátí vypočtenou hodnotu. Pro přepínání mezi ovládáním pomocí akcelerometru a ovládáním pomocí dotykového displeje slouží tlačítko togglesensormode reprezentované proměnnou sensormode_. Pro ovládání pomocí displeje je použit open source joystick s názvem „JoystickView Widget“ od Mobile Anarchy. Díky tomuto joysticku lze ovládat jízdu modelu jedním prstem.

Další prvky jsou tři textová pole. První zobrazuje status připojení k desce Android IOIO, v kolonce je zobrazen nápis „Connected“ pokud je vytvořeno aktivní spojení s deskou Android IOIO. Další dva zobrazují číslo v rozmezí nula až deset. Nula znamená klidový stav, se stoupajícím číslem roste rychlost v ose x nebo míra zatočení kol v ose y. Tyto dva prvky jsou zobrazeny zejména kvůli senzorickému módu pro ověření nastavené rychlosti podle náklonu telefonu.

Poslední dva prvky slouží pro interakce s hardwarovými perifériemi modelu automobilu. Tlačítko s popisem „Lignts“ spíná osvětlení automobilu pomocí tranzistoru IRF530, který je připojen k příslušnému pinu mikroprocesoru. Druhou periferií je optický senzor. Ten slouží k měření rychlosti modelu. Hardwarový senzor zaznamená pouze jednu otočku kola. Pomocí metody ChcekSpeed() vypočteme rychlost jízdy podle průměru kola modelu a času mezi jednotlivými otočkami. Android IOIO neumožní vytvořit programový

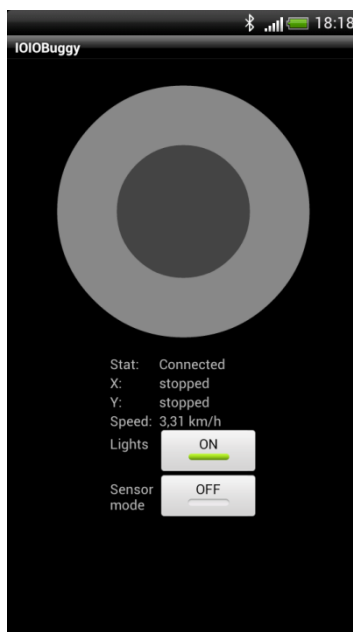
listener, který by spustil určenou část programu v reakci na změnu stavu vstupu mikroprocesoru. Můžeme pouze programově otestovat, zda je odrazová plocha v dosahu senzoru. Proto je metoda `CheckSpeed()` testována ve více částech programu, poté je vytvořen jednoduchý semafor pomocí pomocné proměnné typu boolean s názvem `turnflag`. Semafor zajistí, že nebudeme považovat dvojité načtení odrazové plochy za dvě otočky. Důležité je proto sladění programu se senzorem, zejména velikostí odrazové plochy připevněné ke kolu modelu.

Z ergonomického hlediska je orientace grafiky programu uzamčena pouze do vertikálního zobrazení pomocí následujícího kódu.

```
setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);  
getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);
```

Tato část kódu musí být vložena do inicializační metody `onCreate()`, která je vykonávána při spuštění programu. Tímto způsobem zamezíme překlápění obrazu v závislosti na naklonění telefonu. To se hodí zejména v senzorickém módu, kde by řízením vzdáleného zařízení pomocí akcelerometru docházelo k neustálému překlápění obrazu. Druhý řádek začínající `getWindow().addFlags...` slouží k zamezení ztmavení displeje. Display se bez tohoto kódu sám vypne, pokud není časem vyvolána aktivita, jako je stisknutí displeje nebo některého z tlačítek. V senzorickém módu by došlo k vypnutí displeje a uzamčení obrazovky. To by mělo za následek přerušení běhu aplikace, a tím pádem ztrátu kontroly nad řízeným zařízením.

Výsledný vzhled celé aplikace s výše zmíněnými prvky je zobrazen na obrázku 4.7, který je zachycen sejmutím obrazovky na zařízení EVO3D.

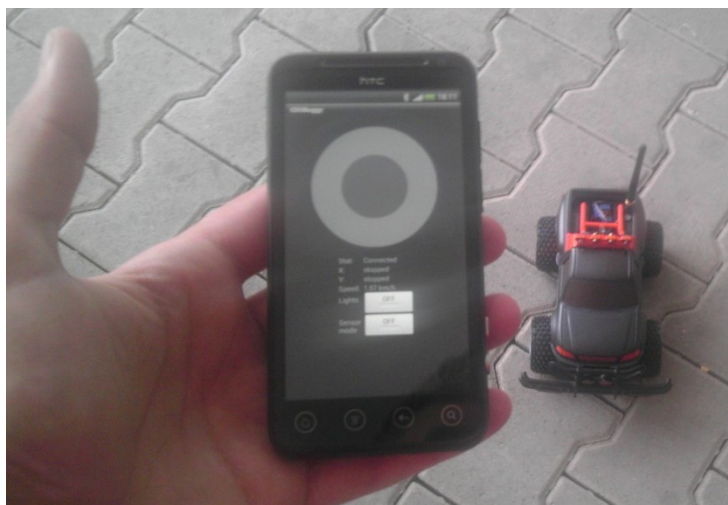


Obrázek 4.7: Výsledné grafické rozhraní aplikace

4.3 Testování

Testování probíhalo zejména na modelu s řídicím obvodem na jeho straně (obr. 4.8), tedy modelu s kompletně obměněnou elektronikou. Při jeho řízení byl subjektivně posuzován vliv zpracování dat akcelerometru na výslednou jízdu. Z testů jednoznačně vyplynula nutnost použití metod zmíněných v kapitole 2.8. S vhodným průměrováním je model mnohem plynuleji ovladatelný a opotřebení pohyblivých částí je zanedbatelné. Dále byl postupně zdokonalován samotný řídicí obvod modelu. Díky provozním testům bylo možné odstranit většinu nedostatků. Krátký dosah byl s použitím antény rozšířen až na 30 m. Díky přidaným ochranným diodám k motoru přestalo docházet k výpadkům komunikace, které nastávaly při rychlé změně otáčení pohonného motoru. Přepracovaný model v některých ohledech předčil původní elektroniku. V modelu je použit původní motor, který je navržen pro maximální vstupní napětí 6V. Jelikož původní výrobce osadil pouze 4.8V baterii, nemohlo být dosaženo maximálního výkonu tohoto motoru. Pro zatáčení byl použit servomotor, který svou přesností dalece předčil původní pružinový systém zatáčení.

Jelikož nebyla předem naměřena odezva původní elektroniky, nemůžeme je tedy přímo porovnat. Nová elektronika byla porovnána s rádiovou soupravou Turnigy 9x. V subjektivním porovnání nelze postřehnout rozdíl mezi stiskem tlačítka ovladače a pohybem modelu. Proto byl proveden video záznam, ze kterého bylo zpoždění odečteno. Jelikož se nejednalo o vysokorychlostní kameru, nemůžeme tento výsledek považovat za přesný. Můžeme však porovnat rádiovou soupravu s naším zařízením bok po boku, a tak zjistit, které je rychlejší. V testu byl měřen čas mezi stiskem tlačítka a pohybem servomotoru. Výsledkem bylo zjištění, že naše řešení je přibližně o 10 ms rychlejší než souprava Turnigy 9x, která disponuje odezvou přibližně 50ms.



Obrázek 4.8: Testovaná sestava

5 Shrnutí a závěr

Cílem této práce bylo popsat a navrhnout kompletní řešení vzdáleného zařízení, řízeného pomocí akcelerometru v mobilním telefonu. V práci bylo názorně popsáno, jak vytvořit řídicí obvod i aplikaci pro mobilní telefon s operačním systémem Android. Výsledkem je hned několik prototypů, na kterých lze získané vědomosti dobře ověřit. Tato práce je dobrým důkazem prolínání všech oborů s oborem informatiky a poukazuje na stále nové možnosti využití mobilních zařízení.

Osobní přínos vidím, jak ve zdokonalení svých programátorských dovedností, tak i dovedností spojených s tvorbou elektronických obvodů. Práce mě nutí dále přemýšlet, k jakým dalším účelům lze mobilní telefon využít.

Tento systém řízení pomocí akcelerometru v mobilním telefonu byl shledán jako ideální alternativa pro bezdrátové soupravy používané pro řízení modelů na dálkové ovládání. V tomto ohledu lze dále rozvíjet mobilní aplikaci tak, aby bylo možné ovládat nejrůznější typy pozemních i leteckých modelů. Dále se chci věnovat zejména rozvíjení telemetrie, která umožní přenášet informace ze senzorů modelu na display mobilního telefonu. Výzvou bude sledovat, zda se toto řešení stane populární, a zda se z něj může stát plnohodnotný produkt.

Citovaná literatura

- [1] **Webnesters Pvt.Ltd.** Piezoelectric Transducer. *Instrumentation Today*. [Online] 27. čerčenec 2011. [Citace: 21. Únor 3013.] <http://www.instrumentationtoday.com/piezoelectric-transducer/2011/07/>.
- [2] **Webnesters Pvt.Ltd.** MEMS Accelerometer. *Instrumentation Today*. [Online] 17. Srpen 2011. [Citace: 21. Únor 3013.] <http://www.instrumentationtoday.com/mems-accelerometer/2011/08/>.
- [3] **Analog Devices, Inc.** Dual Axis, Low g, Fully Integrated Accelerometers. *Analog Devices*. [Online] 1999. [Citace: 22. Únor 2013.] <http://www.analog.com/library/analogDialogue/archives/33-01/accel/adxl202.pdf>.
- [4] **Wi-Fi Alliance.** Wi-Fi Alliance. *Wi-Fi Alliance*. [Online] 2013. [Citace: 20. Březen 2013.] <http://www.wi-fi.org/>.
- [5] **Bluetooth SIG.** Welcome to Bluetooth. *Bluetooth special interest group*. [Online] 2013. [Citace: 20. Březen 2013.] <https://www.bluetooth.org/>.
- [6] **Open Source Hardware Association.** OSHW Definition. *Open Source Hardware Association*. [Online] [Citace: 25. Březen 2013.] <http://www.oshwa.org/definition/>.
- [7] **Arduino.** ArduinoBoardADK. *Arduino*. [Online] 28. Leden 2013. [Citace: 25. Březen 2013.] <http://arduino.cc/en/Main/ArduinoBoardADK>.
- [8] **Ben-Tsvi, Ytai.** home · ytai/ioio Wiki · GitHub. *GitHub*. [Online] GitHub, Inc, 2013. [Citace: 09. Duben 2013.] <https://github.com/ytai/ioio/wiki>.
- [9] **Monk, Simon.** *Making Android Acestories with IOIO*. Sebastopol : O'REILLY, 2012. ISBN 978-1-449-32328-8.

Seznam obrázků

Obrázek 2.1: Akcelerometr z mechanických částí vlevo a akcelerometr s krystalem vpravo	2
Obrázek 2.2: Kapacitní část MEMS akcelerometru.....	3
Obrázek 2.3: Akcelerometr ADXL202 pod mikroskopem [3].....	3
Obrázek 2.4: a) Pohyb měřený akcelerometrem b) Gyroskopický pohyb	4
Obrázek 2.5: Přípravek pro testování akcelerometru	6
Obrázek 2.6: Linearita přípravku	8
Obrázek 2.7: Linearita Evo3D osa X	9
Obrázek 2.8: Linearita Desire osa X	9
Obrázek 2.9: Hladinový test osy x zařízení Evo3D	10
Obrázek 2.10: Rozptyly ve všech hladinách	10
Obrázek 2.11: Reakce na rychlé zastavení osy X zařízení Evo3D	13
Obrázek 2.12: Reakce na rychlé zastavení osy X zařízení Desire	13
Obrázek 2.13: Porovnání rychlostí v jednotlivých módech	15
Obrázek 2.14: Porovnání reálného pohybu s daty akcelerometru.....	16
Obrázek 2.15: Porovnání aritmetického průměru podle množství vzorků.....	16
Obrázek 2.16: Porovnání mediánového vyhlazení dle množství vzorků	17
Obrázek 2.17: Porovnání reálného signálu se signálem upraveným.....	17
Obrázek 3.1: Kartáčkový stejnosměrný motor.....	23
Obrázek 3.2: Bezkartáčkový stejnosměrný motor a jeho řídicí obvod	24
Obrázek 3.3: Základní krokový motor a jeho řídicí obvod	24
Obrázek 3.4: Servomotor	25
Obrázek 3.5: Relé s ochranou diodou	25
Obrázek 3.6: Tranzistor MOSFET	26
Obrázek 3.7: Arduino ADK R3 přední a zadní strana [7].....	28
Obrázek 3.8: Android IOIO verze 1.8.....	29
Obrázek 3.9: Klon Arduino UNO vlevo, Bluetooth modul vpravo	30

Obrázek 3.10: Vývojový kit Stellaris LaunchPad LM4F120.....	31
Obrázek 4.1: Původní model s ovladačem.....	33
Obrázek 4.2: Upravený ovladač.....	33
Obrázek 4.3 Řídící obvod modelu	35
Obrázek 4.4: Optický spínač sledující otáčení kola	35
Obrázek 4.5: Souprava Turnigy 9x vlevo a souprava s Android IOIO vpravo	37
Obrázek 4.6: Návrh plošného spoje univerzální nástavby pro Android IOIO.	38
Obrázek 4.7: Výsledné grafické rozhraní aplikace	41
Obrázek 4.8: Testovaná sestava	42

Seznam tabulek

Tabulka 2.1: Detail testovaných zařízení	7
Tabulka 2.2: Hladinový test Evo3D	11
Tabulka 2.3: Hladinový test Desire.....	11
Tabulka 2.4: Průměrné rychlosti odečítání	14
Tabulka 4.1: Cenová kalkulace vysílačů.....	36

Seznam příloh

Příloha.A: Typy senzorů třídy Sensor	I
Příloha.B: Zpoždění vložené systémem Android	I
Příloha.C: Linearita Evo3D a Desire v osách x, z.	II
Příloha.D: Reakce na rychlé zastavení Evo3D a Desire v osách x, z.	III
Příloha.E: Obvodové řešení ovladače rozšířeného o Bluetooth	IV
Příloha.F: Kompletní obvodové řešení modelu s bluetooth konektivitou.....	V
Příloha.G: Návrh nadstavby pro Android IOIO sloužící jako univerzální ovladač	VI

Přílohy

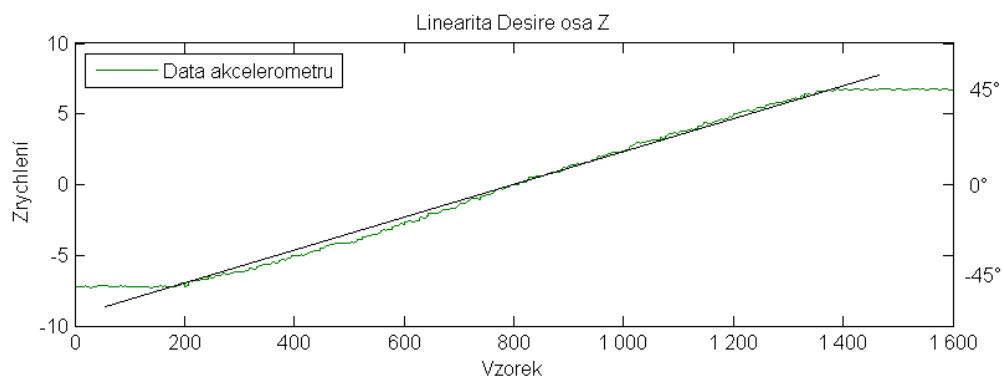
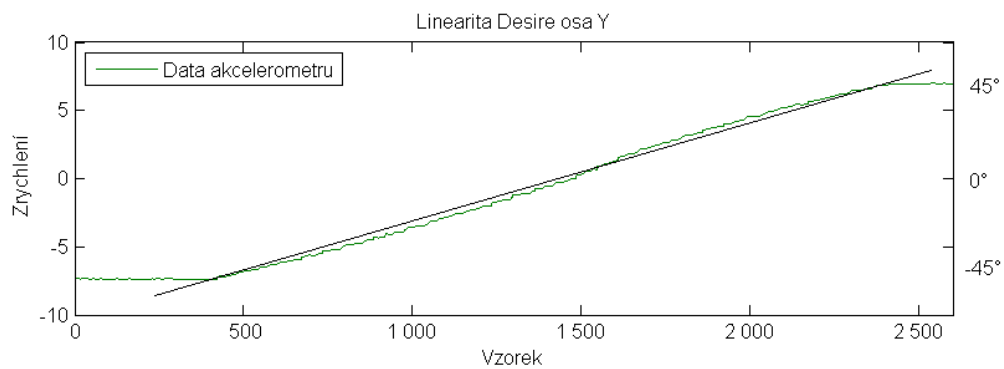
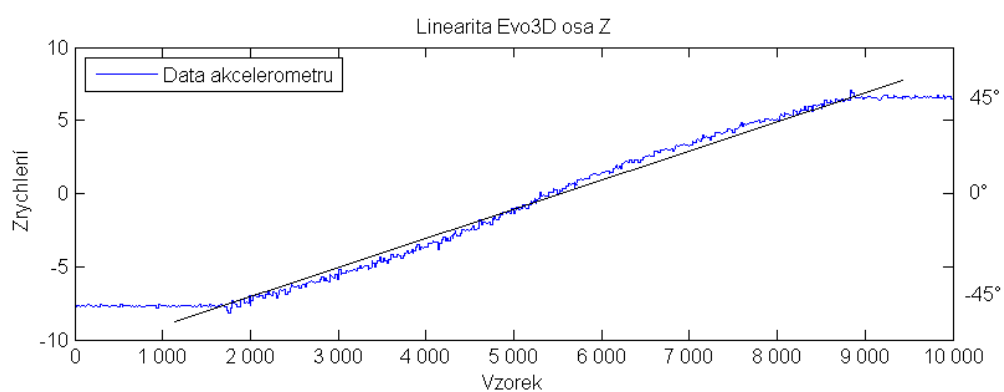
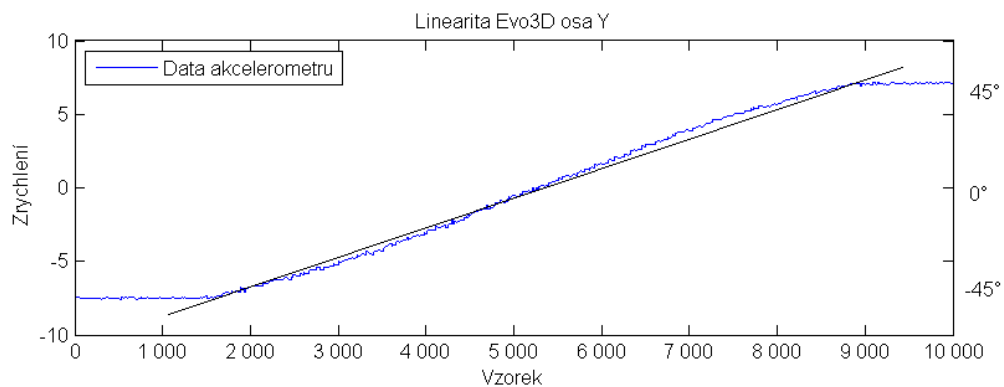
Príloha.A: Typy senzorů třídy Sensor

Konstanta	Popis
TYPE_ACCELEROMETER	Senzor typu akcelerometr
TYPE_ALL	Popisuje všechny senzory
TYPE_AMBIENT_TEMPERATURE	Senzor okolní teploty
TYPE_GRAVITY	Senzor gravitace
TYPE_GYROSCOPE	Senzor typu gyroskop
TYPE_LIGHT	Senzor osvětlení
TYPE_LINEAR_ACCELERATION	Senzor lineárního zrychlení
TYPE_MAGNETIC_FIELD	Senzor magnetického pole
TYPE_ORIENTATION	Senzor orientace, natočení
TYPE_PRESSURE	Senzor tlaku
TYPE_PROXIMITY	Senzor vzdálenosti
TYPE_RELATIVE_HUMIDITY	Senzor vlhkosti
TYPE_ROTATION_VECTOR	Měření rotačního vektoru
TYPE_TEMPERATURE	Senzor teploty

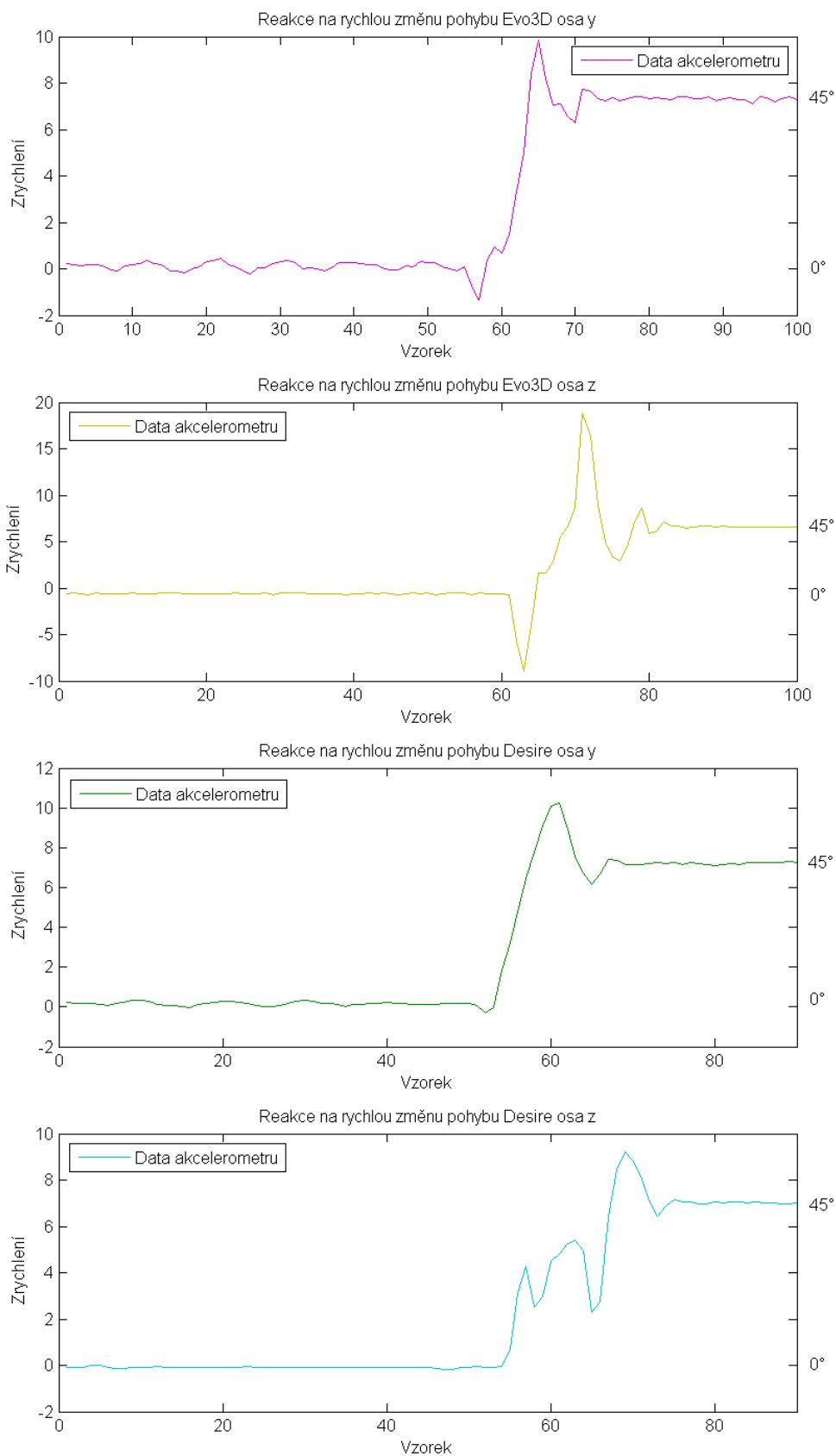
Príloha.B: Zpoždění vložené systémem Android

Typ odezvy	Popis	Relativní rychlost
SENSOR_DELAY_FASTEST	Odečítání dat nejrychleji jak je možno	Nejrychlejší
SENSOR_DELAY_GAME	Rychlost vhodná pro hry	Rychlý
SENSOR_DELAY_UI	Pro ovládání uživatelského rozhraní	Pomalejší
SENSOR_DELAY_NORMAL	Vhodné pro změnu orientace obrazovky	Nejpomalejší

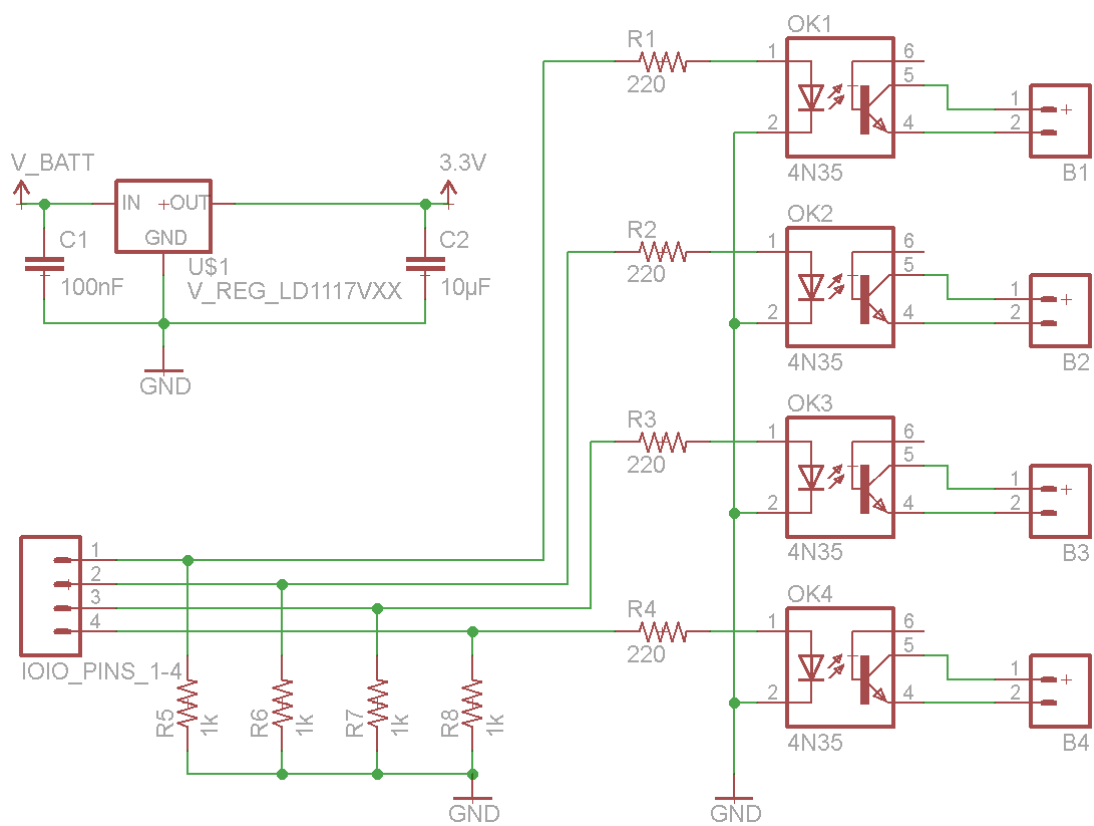
Příloha.C: Linearita Evo3D a Desire v osách x, z.



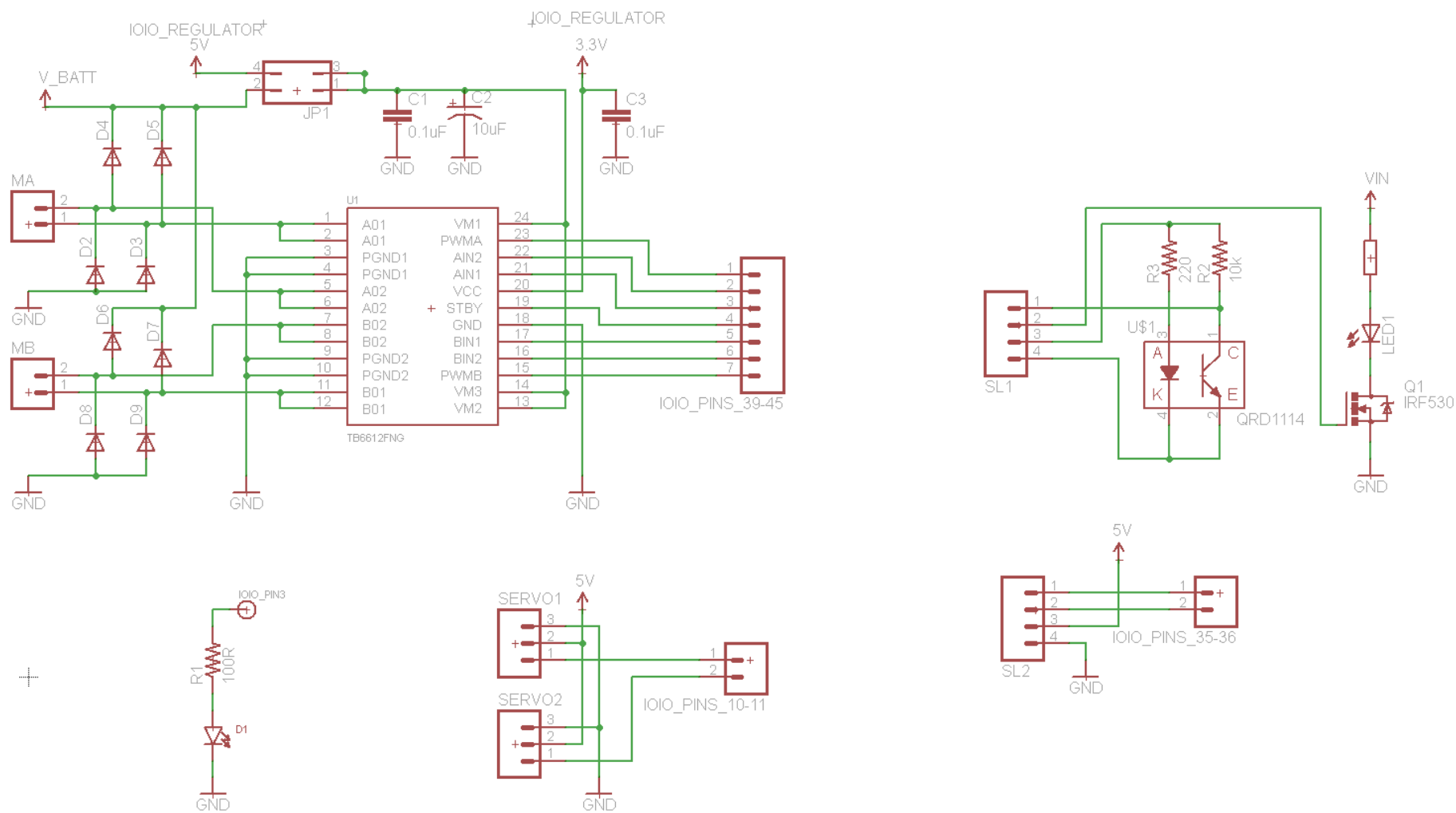
Příloha.D: Reakce na rychlé zastavení Evo3D a Desire v osách x, z.



Příloha.E: Obvodové řešení ovladače rozšířeného o Bluetooth



Příloha.F: Kompletní obvodové řešení modelu s bluetooth konektivitou



Příloha.G: Návrh nadstavby pro Android IOIO sloužící jako univerzální ovladač

